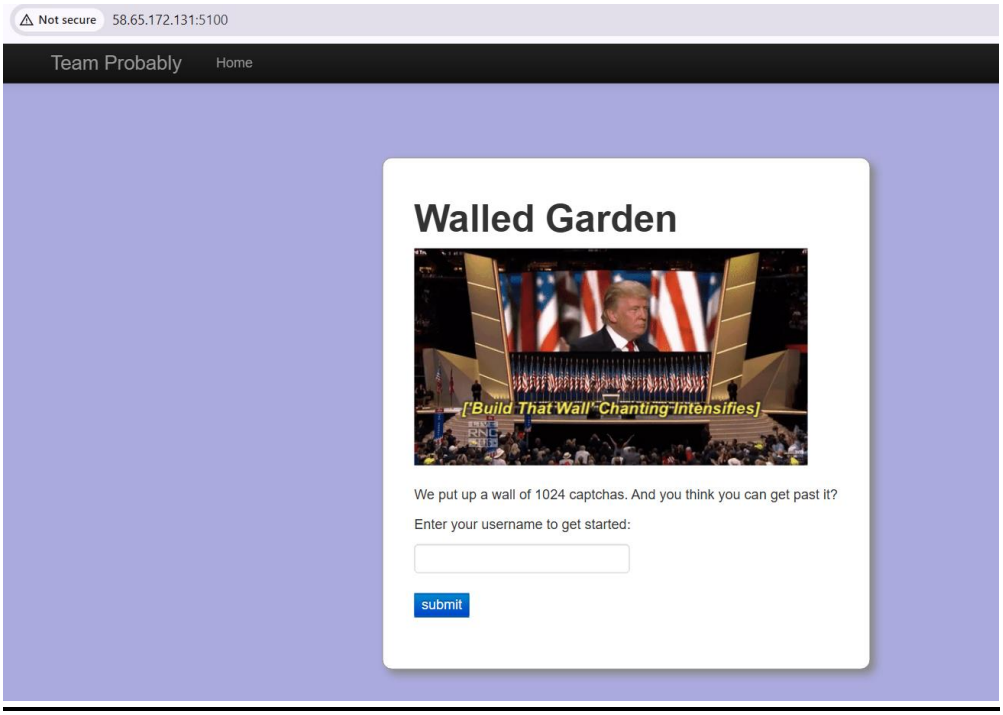
TRILLIUM CTF

Zakyas Ali

Task 01-05 Write-up

Task 1



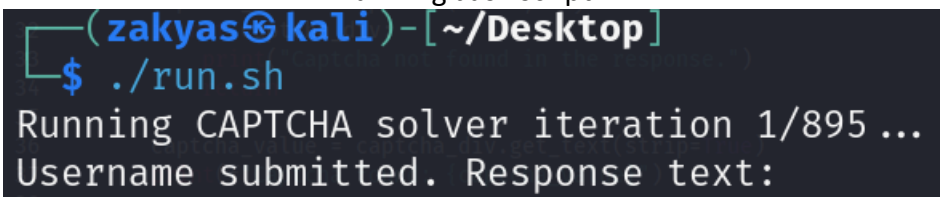
creating a python script that can read and post captcha single time:

```python
import requests
from bs4 import BeautifulSoup

# URL for the CAPTCHA challenge
BASE_URL = "http://58.65.172.131:5100/"  # Replace with the actual URL
CAPTCHA_ENDPOINT = f"{BASE_URL}/index.php"

# Username and max captchas
USERNAME = "admin"
MAX_CAPTCHAS = 1024

def solve_captchas():
    session = requests.Session()  # Maintain the session to handle cookies, etc.

    # Initial request to get the first CAPTCHA
    payload = {"name": USERNAME}
    response = session.get(CAPTCHA_ENDPOINT, params=payload)

    if response.status_code != 200:
        print("Failed to submit username. Status code:", response.status_code)
        return

    print("Username submitted. Response text:\n", response.text)

    captcha_count = 0

    while captcha_count < MAX_CAPTCHAS:
        soup = BeautifulSoup(response.text, 'html.parser')

        # Extract the CAPTCHA value
        captcha_div = soup.find("div", style=lambda x: x and "font-family" in x)
        if not captcha_div:
            print("Captcha not found in the response.")
            return

        captcha_value = captcha_div.get_text(strip=True)
        print(f"Captcha found: {captcha_value}")

        # Prepare the form submission payload
        payload["captcha"] = captcha_value
        submit_response = session.get(CAPTCHA_ENDPOINT, params=payload)

        if submit_response.status_code != 200:
            print("Failed to submit the CAPTCHA. Status code:", submit_response.status_code)
            return

        # Check for success message
        if "Captcha correct!" in submit_response.text:
            captcha_count += 1
            print(f"Captcha solved! Solved {captcha_count} captchas.")

            # If all captchas are solved, look for the flag in the final response
            if "1024 captchas" in submit_response.text:
                print("Successfully solved all captchas! Check for the flag.")
                print("Final Response HTML:")
                print(submit_response.text)  # This should contain the flag
                break
        else:
            print("Captcha solving failed. Response text:\n", submit_response.text)
            break

        # Update the response to continue solving the next CAPTCHA
        response = submit_response

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

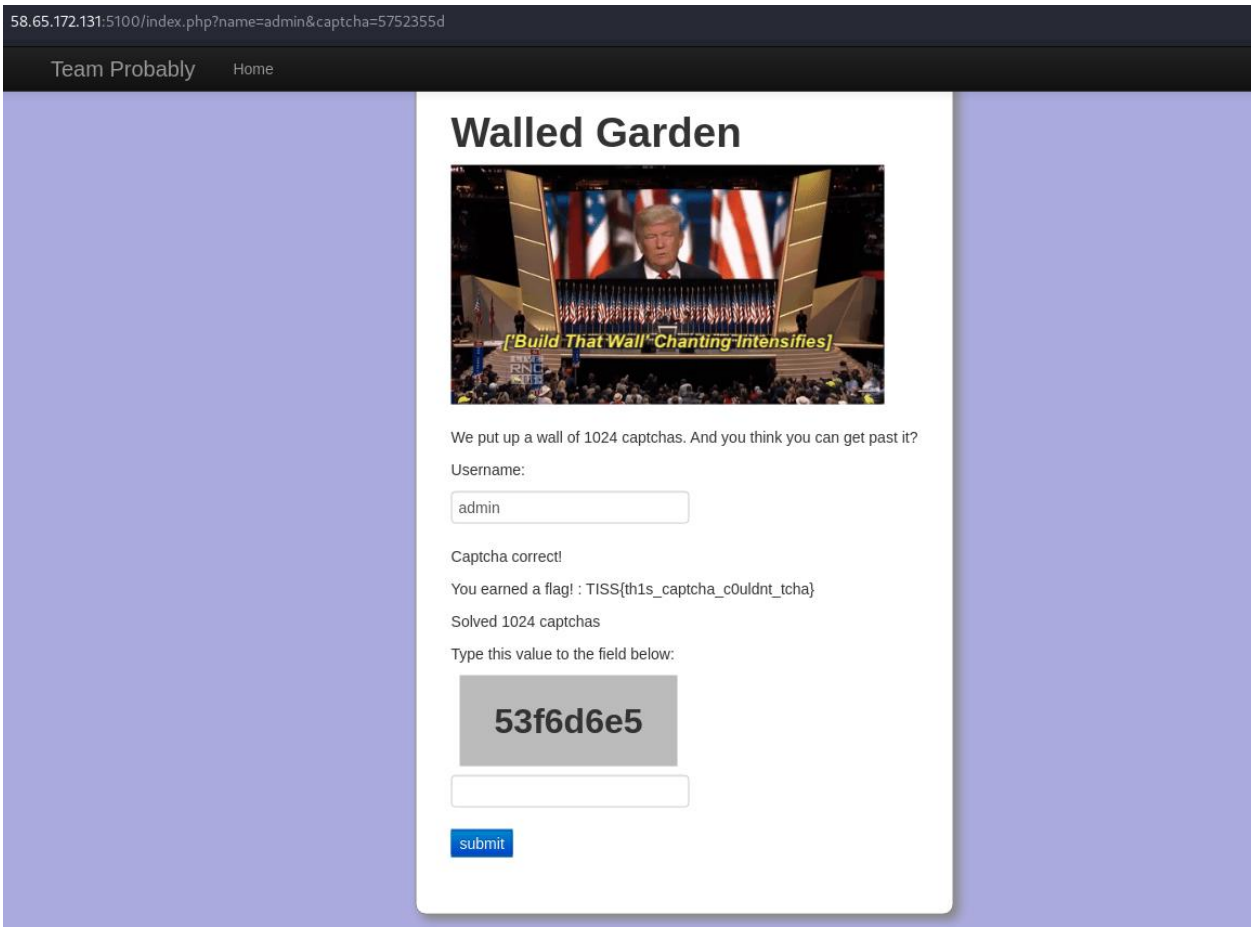
creating a bash script that will run python script 1024 time:

```
┌──(zakyas㉿kali)-[~/Desktop]
└─$ cat run.sh
#!/bin/bash

# Number of times to run the Python script
NUM_RUNS=895

# Loop to run the Python script the specified number of times
for ((i=1; i<=NUM_RUNS; i++))
do
    echo "Running CAPTCHA solver iteration $i/$NUM_RUNS..."
    python3 capt.py  # Replace with the correct path if needed
    echo "-----------------------------------"
done
```

running bash script:



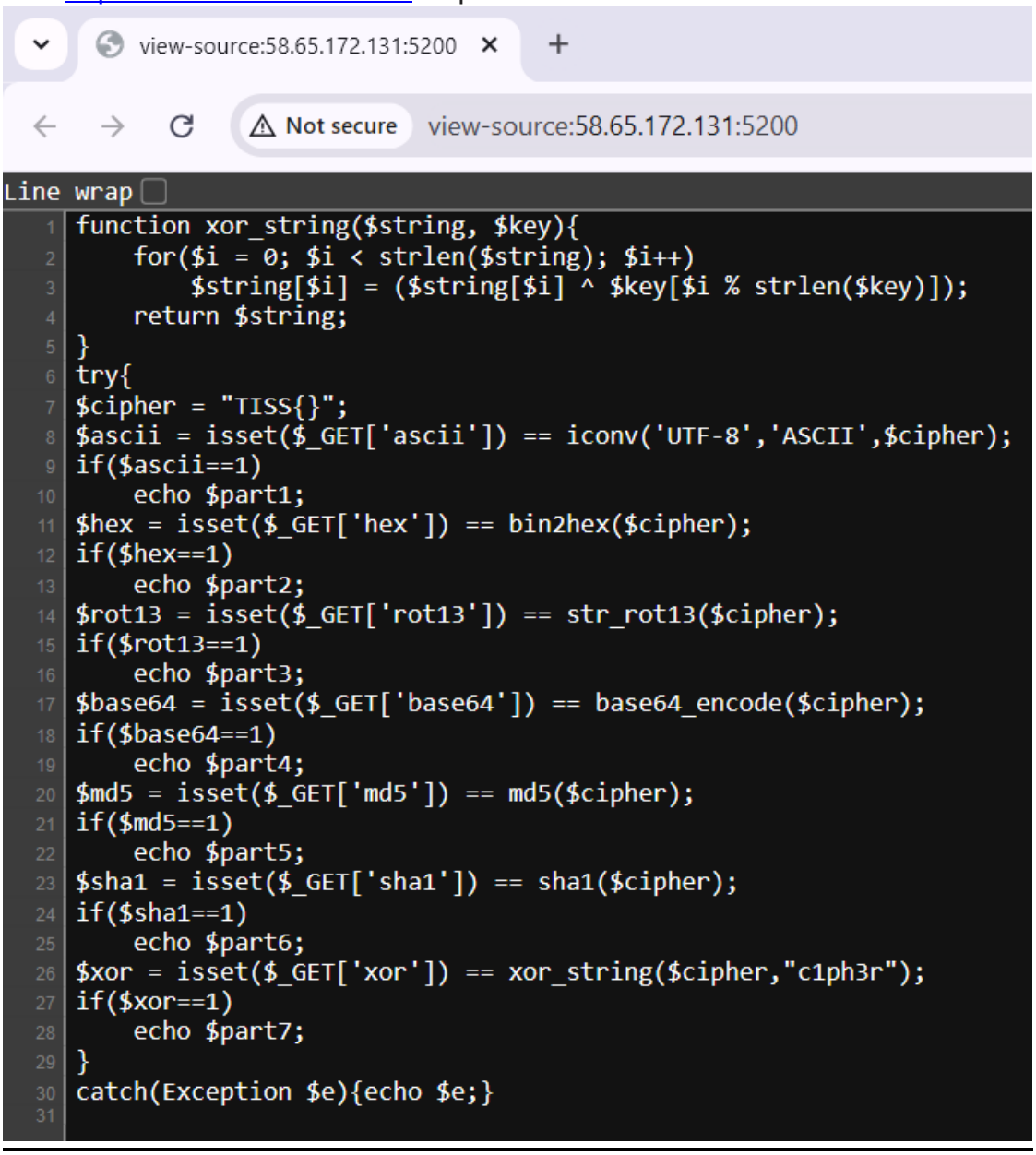
after 1024 times captcha submission, we get flag on web page:



Flag: TISS{th1s_captcha_c0uldnt_tcha}

Task 2

Link: http://58.65.172.131:5200/ - cipher




```php
function xor_string($string, $key){
    for($i = 0; $i < strlen($string); $i++)
        $string[$i] = ($string[$i] ^ $key[$i % strlen($key)]);
    return $string;
}
try{
$cipher = "TISS{}";
$ascii = isset($_GET['ascii']) == iconv('UTF-8','ASCII',$cipher);
if($ascii==1)
    echo $part1;
$hex = isset($_GET['hex']) == bin2hex($cipher);
if($hex==1)
    echo $part2;
$rot13 = isset($_GET['rot13']) == str_rot13($cipher);
if($rot13==1)
    echo $part3;
$base64 = isset($_GET['base64']) == base64_encode($cipher);
if($base64==1)
    echo $part4;
$md5 = isset($_GET['md5']) == md5($cipher);
if($md5==1)
    echo $part5;
$sha1 = isset($_GET['sha1']) == sha1($cipher);
if($sha1==1)
    echo $part6;
$xor = isset($_GET['xor']) == xor_string($cipher,"c1ph3r");
if($xor==1)
    echo $part7;
}
catch(Exception $e){echo $e;}
```

by analyzing this php code, requesting url with correct variables value will echo flag parts. Variables values are set with different encryption methods applied.

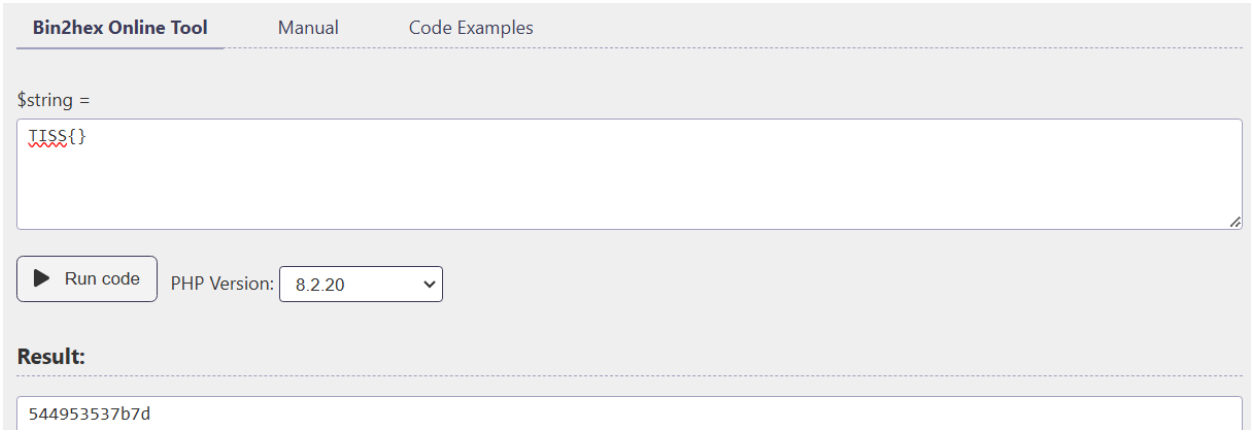
Part 1:

```
┌──(zakyas㉿kali)-[~/Desktop]
└─$ curl "http://58.65.172.131:5200/?ascii=TISS"
function xor_string($string, $key){
    for($i = 0; $i < strlen($string); $i++)
        $string[$i] = ($string[$i] ^ $key[$i % strlen($key)]);
    return $string;
}
try{
$cipher = "TISS{}";
$ascii = isset($_GET['ascii']) == iconv('UTF-8','ASCII',$cipher);
if($ascii==1)
    echo $part1;
$hex = isset($_GET['hex']) == bin2hex($cipher);
if($hex==1)
    echo $part2;
$rot13 = isset($_GET['rot13']) == str_rot13($cipher);
if($rot13==1)
    echo $part3;
$base64 = isset($_GET['base64']) == base64_encode($cipher);
if($base64==1)
    echo $part4;
$md5 = isset($_GET['md5']) == md5($cipher);
if($md5==1)
    echo $part5;
$sha1 = isset($_GET['sha1']) == sha1($cipher);
if($sha1==1)
    echo $part6;
$xor = isset($_GET['xor']) == xor_string($cipher,"c1ph3r");
if($xor==1)
    echo $part7;
}
catch(Exception $e){echo $e;}
TISS{
```

flag: TISS{

Part 2:

encoding $cipher to bin2hex:

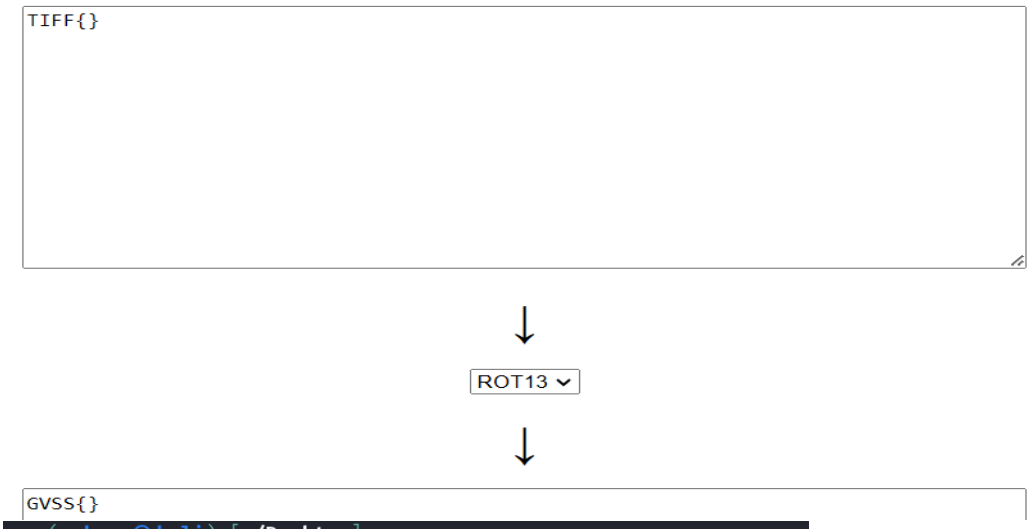



```
┌──(zakyas㉿kali)-[~/Desktop]
└─$ curl "http://58.65.172.131:5200/?hex=544953537b7d"
function xor_string($string, $key){
    for($i = 0; $i < strlen($string); $i++)
        $string[$i] = ($string[$i] ^ $key[$i % strlen($key)]);
    return $string;
}
try{
$cipher = "TISS{}";
$ascii = isset($_GET['ascii']) == iconv('UTF-8','ASCII',$cipher);
if($ascii==1)
    echo $part1;
$hex = isset($_GET['hex']) == bin2hex($cipher);
if($hex==1)
    echo $part2;
$rot13 = isset($_GET['rot13']) == str_rot13($cipher);
if($rot13==1)
    echo $part3;
$base64 = isset($_GET['base64']) == base64_encode($cipher);
if($base64==1)
    echo $part4;
$md5 = isset($_GET['md5']) == md5($cipher);
if($md5==1)
    echo $part5;
$sha1 = isset($_GET['sha1']) == sha1($cipher);
if($sha1==1)
    echo $part6;
$xor = isset($_GET['xor']) == xor_string($cipher,"c1ph3r");
if($xor==1)
    echo $part7;
}
catch(Exception $e){echo $e;}
s0m3b0dy_
```

flag: s0m3b0dy_

Part 3:

encoding $cipher to ROT13:

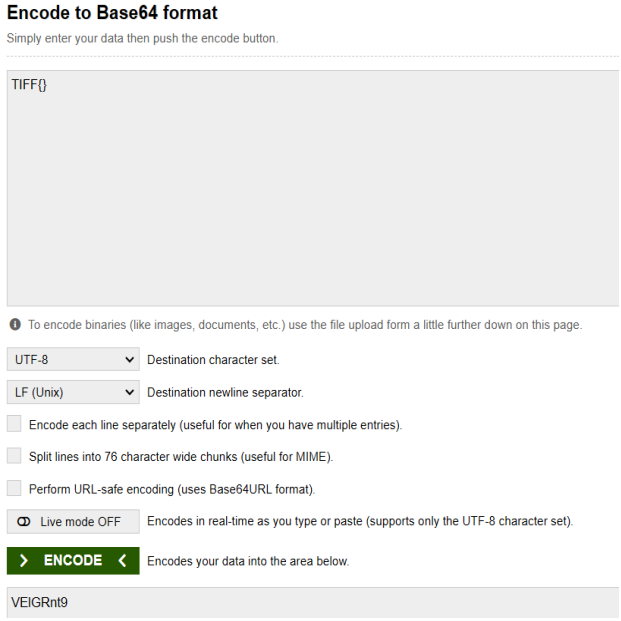



```
┌──(zakyas㉿kali)-[~/Desktop]
└─$ curl "http://58.65.172.131:5200/?rot13=GVFF"

function xor_string($string, $key){
    for($i = 0; $i < strlen($string); $i++)
        $string[$i] = ($string[$i] ^ $key[$i % strlen($key)]);
    return $string;
}
try{
$cipher = "TISS{}";
$ascii = isset($_GET['ascii']) == iconv('UTF-8','ASCII',$cipher);
if($ascii==1)
    echo $part1;
$hex = isset($_GET['hex']) == bin2hex($cipher);
if($hex==1)
    echo $part2;
$rot13 = isset($_GET['rot13']) == str_rot13($cipher);
if($rot13==1)
    echo $part3;
$base64 = isset($_GET['base64']) == base64_encode($cipher);
if($base64==1)
    echo $part4;
$md5 = isset($_GET['md5']) == md5($cipher);
if($md5==1)
    echo $part5;
$sha1 = isset($_GET['sha1']) == sha1($cipher);
if($sha1==1)
    echo $part6;
$xor = isset($_GET['xor']) == xor_string($cipher,"c1ph3r");
if($xor==1)
    echo $part7;
}
catch(Exception $e){echo $e;}
w1ll_b3_4bl3_t0
```

flag: w1ll_b3_4bl3_t0

Part 4:

encoding $cipher to base 64:




```
┌──(zakyas㉿kali)-[~/Desktop]
└─$ curl "http://58.65.172.131:5200/?base64=VElTU3t9"

function xor_string($string, $key){
    for($i = 0; $i < strlen($string); $i++)
        $string[$i] = ($string[$i] ^ $key[$i % strlen($key)]);
    return $string;
}
try{
$cipher = "TISS{}";
$ascii = isset($_GET['ascii']) == iconv('UTF-8','ASCII',$cipher);
if($ascii==1)
    echo $part1;
$hex = isset($_GET['hex']) == bin2hex($cipher);
if($hex==1)
    echo $part2;
$rot13 = isset($_GET['rot13']) == str_rot13($cipher);
if($rot13==1)
    echo $part3;
$base64 = isset($_GET['base64']) == base64_encode($cipher);
if($base64==1)
    echo $part4;
$md5 = isset($_GET['md5']) == md5($cipher);
if($md5==1)
    echo $part5;
$sha1 = isset($_GET['sha1']) == sha1($cipher);
if($sha1==1)
    echo $part6;
$xor = isset($_GET['xor']) == xor_string($cipher,"c1ph3r");
if($xor==1)
    echo $part7;
}
catch(Exception $e){echo $e;}
_0v3rc0m3_
```

flag: _0v3rc0m3_

Part 5:

encoding $cipher to md5 hash:

Your String	TIFF{}
MD5 Hash	15fe5420eba18cc2306ae658d4617534 Copy

```
┌──(zakyas㉿kali)-[~/Desktop]
└─$ curl "http://58.65.172.131:5200/?md5=1567c640c585d43b03d450c247e0ad38"

function xor_string($string, $key){
    for($i = 0; $i < strlen($string); $i++)
        $string[$i] = ($string[$i] ^ $key[$i % strlen($key)]);
    return $string;
}
try{
$cipher = "TISS{}";
$ascii = isset($_GET['ascii']) == iconv('UTF-8','ASCII',$cipher);
if($ascii==1)
    echo $part1;
$hex = isset($_GET['hex']) == bin2hex($cipher);
if($hex==1)
    echo $part2;
$rot13 = isset($_GET['rot13']) == str_rot13($cipher);
if($rot13==1)
    echo $part3;
$base64 = isset($_GET['base64']) == base64_encode($cipher);
if($base64==1)
    echo $part4;
$md5 = isset($_GET['md5']) == md5($cipher);
if($md5==1)
    echo $part5;
$sha1 = isset($_GET['sha1']) == sha1($cipher);
if($sha1==1)
    echo $part6;
$xor = isset($_GET['xor']) == xor_string($cipher,"c1ph3r");
if($xor==1)
    echo $part7;
}
catch(Exception $e){echo $e;}
4ny_3ncryp7
```

flag: 4ny_3ncryp7

Part 6:

encoding $cipher to sha1 hash:

Your String	TIFF{}
MD5 Hash	15fe5420eba18cc2306ae658d4617534 Copy
SHA1 Hash	81928b922e7088078b3f3944db71e2d8fe96cddb Copy

```
┌──(zakyas㉿kali)-[~/Desktop]
└─$ curl "http://58.65.172.131:5200/?sha1=f2784ae6dfe57031ca7912bb7005bede5edb853f"

function xor_string($string, $key){
    for($i = 0; $i < strlen($string); $i++)
        $string[$i] = ($string[$i] ^ $key[$i % strlen($key)]);
    return $string;
}
try{
$cipher = "TISS{}";
$ascii = isset($_GET['ascii']) == iconv('UTF-8','ASCII',$cipher);
if($ascii==1)
    echo $part1;
$hex = isset($_GET['hex']) == bin2hex($cipher);
if($hex==1)
    echo $part2;
$rot13 = isset($_GET['rot13']) == str_rot13($cipher);
if($rot13==1)
    echo $part3;
$base64 = isset($_GET['base64']) == base64_encode($cipher);
if($base64==1)
    echo $part4;
$md5 = isset($_GET['md5']) == md5($cipher);
if($md5==1)
    echo $part5;
$sha1 = isset($_GET['sha1']) == sha1($cipher);
if($sha1==1)
    echo $part6;
$xor = isset($_GET['xor']) == xor_string($cipher,"c1ph3r");
if($xor==1)
    echo $part7;
}
catch(Exception $e){echo $e;}
10n_73chn1qu3
```

flag: 10n_73chn1qu3

Part 7:

running given function:

```
┌──(zakyas㉿kali)-[~/Desktop]
└─$ cat xor.php
<?php
function xor_string($string, $key){
    for($i = 0; $i < strlen($string); $i++){
        $string[$i] = $string[$i] ^ $key[$i % strlen($key)];
    }
    return $string;
}

echo xor_string("TISS{}", "c1ph3r");

┌──(zakyas㉿kali)-[~/Desktop]
└─$ php xor.php
7x#;H
```

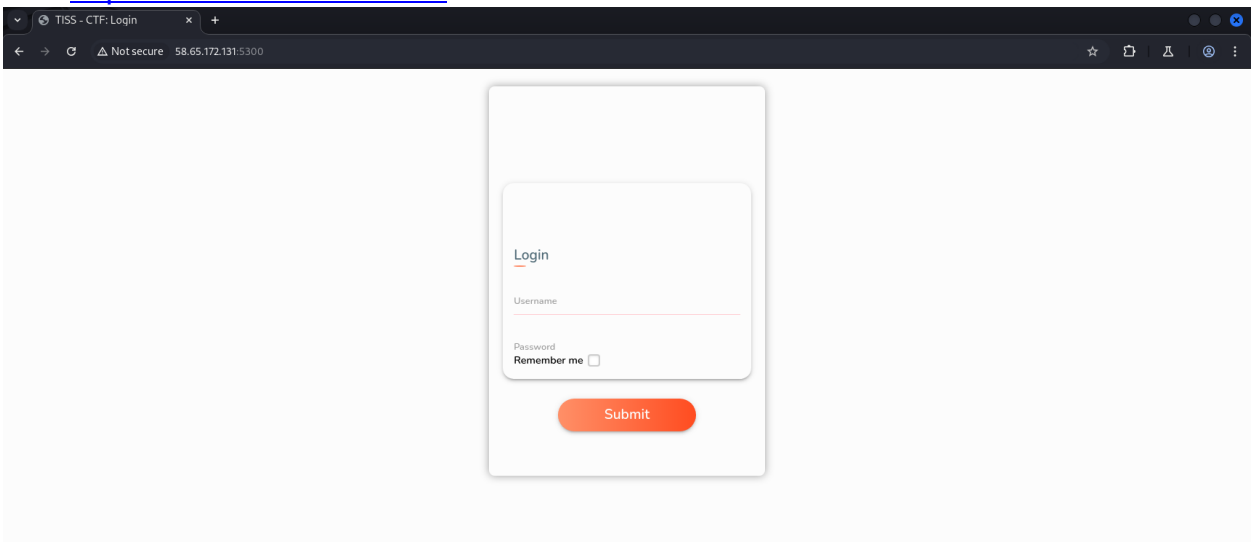
```
┌──(zakyas㉿kali)-[~/Desktop]
└─$ curl "http://58.65.172.131:5200/?xor=7x#;H"

function xor_string($string, $key){
    for($i = 0; $i < strlen($string); $i++)
        $string[$i] = ($string[$i] ^ $key[$i % strlen($key)]);
    return $string;
}
try{
$cipher = "TISS{}";
$ascii = isset($_GET['ascii']) == iconv('UTF-8','ASCII',$cipher
if($ascii==1)
    echo $part1;
$hex = isset($_GET['hex']) == bin2hex($cipher);
if($hex==1)
    echo $part2;
$rot13 = isset($_GET['rot13']) == str_rot13($cipher);
if($rot13==1)
    echo $part3;
$base64 = isset($_GET['base64']) == base64_encode($cipher);
if($base64==1)
    echo $part4;
$md5 = isset($_GET['md5']) == md5($cipher);
if($md5==1)
    echo $part5;
$sha1 = isset($_GET['sha1']) == sha1($cipher);
if($sha1==1)
    echo $part6;
$xor = isset($_GET['xor']) == xor_string($cipher,"c1ph3r");
if($xor==1)
    echo $part7;
}
catch(Exception $e){echo $e;}
_y0u_us3}
```

flag: _y0u_us3}

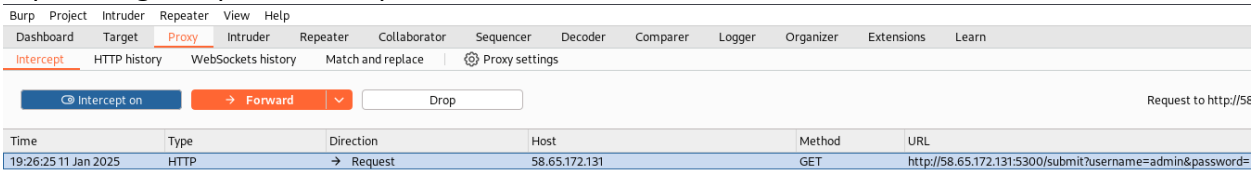
full flag:
TISS{s0m3b0dy_w1ll_b3_4bl3_t0_0v3rc0m3_4ny_3ncryp710n_73chn1qu3_y0u_us3}

Task 3

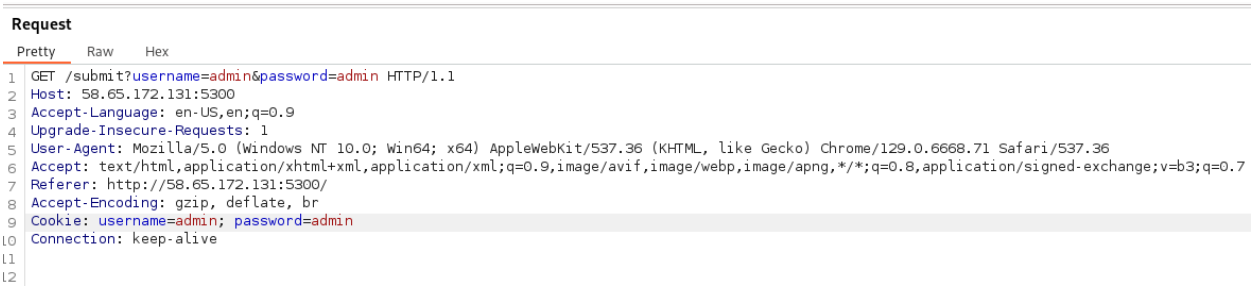
Link: http://58.65.172.131:5300/ - cookie



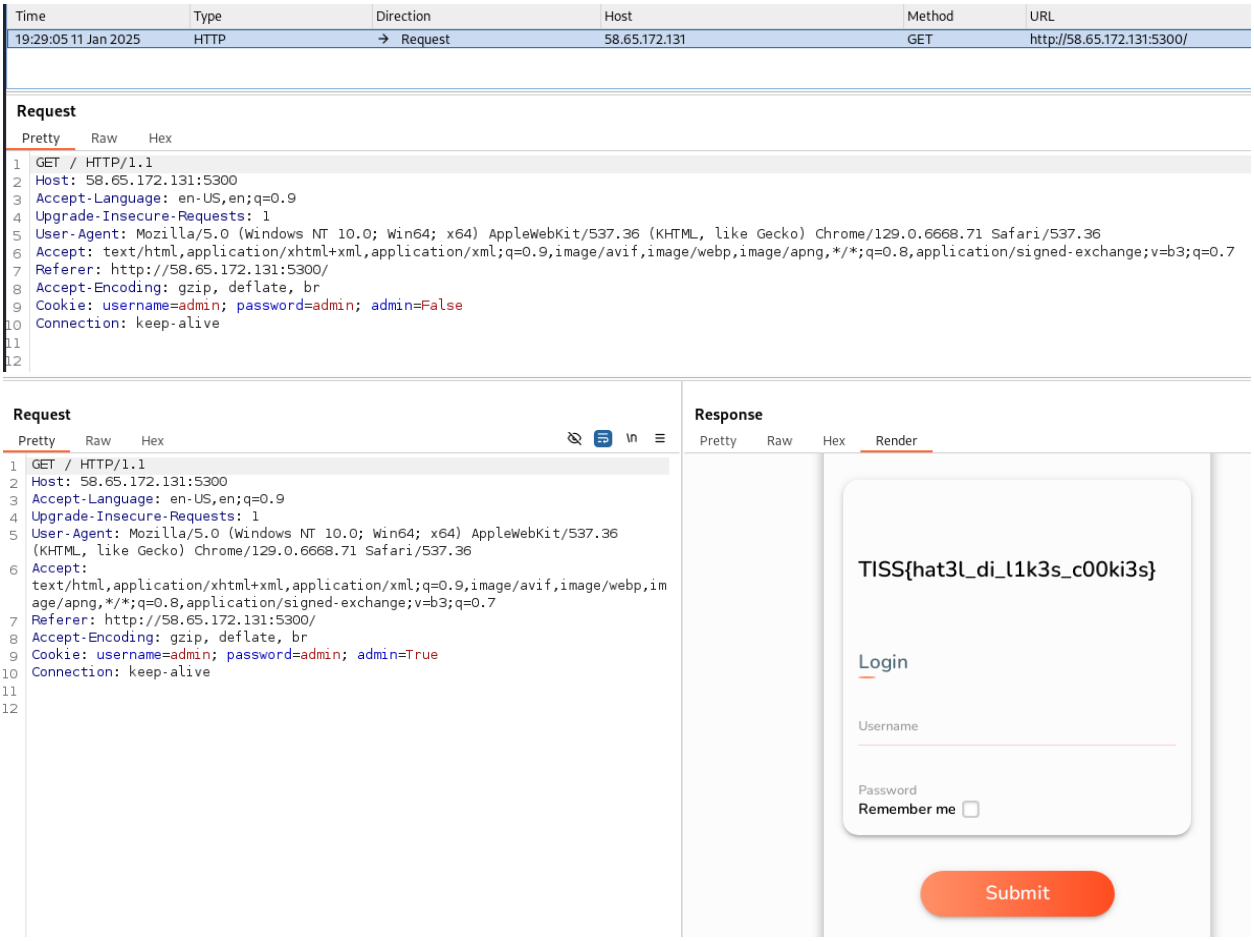
capture login request via burp suite:



modifying GET & cookie headers 'username' & 'password' fields to common "**admin**"



In second request we get additional 'admin' field in cookie which by changing it to true gives flag on webpage:

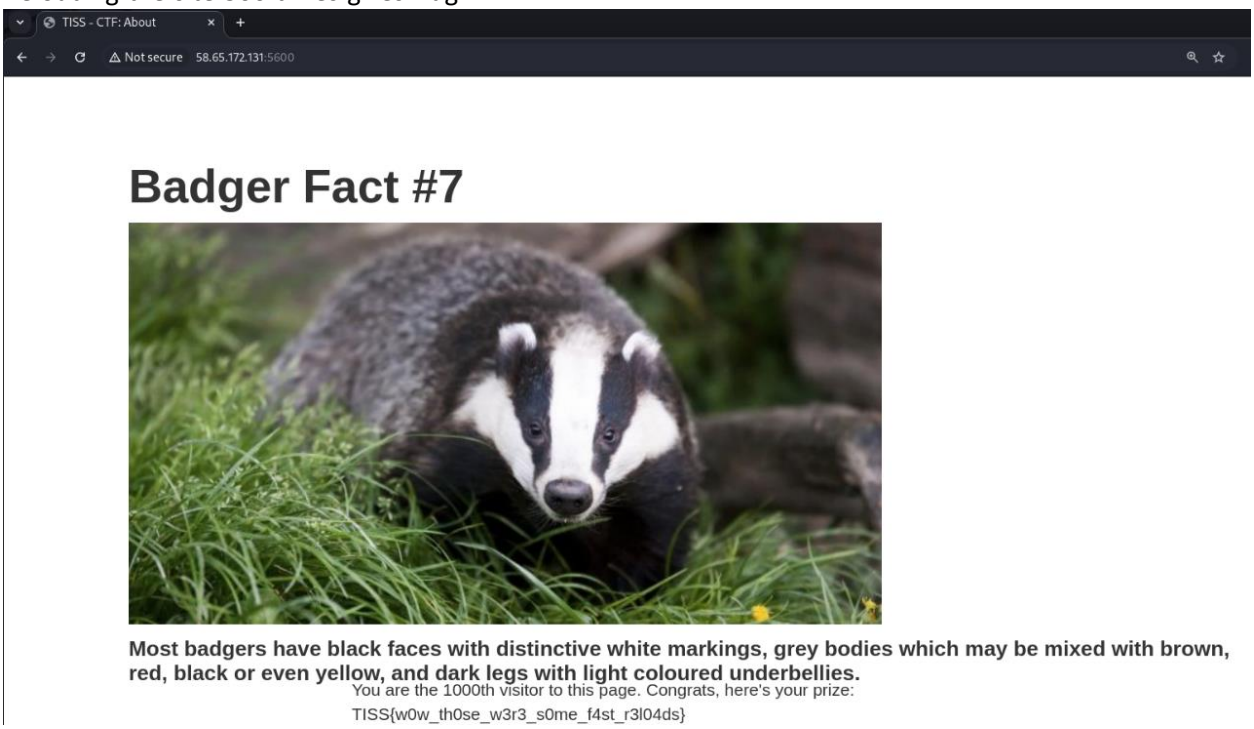


flag: TISS{hat3l_di_l1k3s_c00ki3s}

Task 5

Link: http://58.65.172.131:5600/ - reload

Reloading the site 500 times gives flag:



flag: TISS{w0w_th0se_w3r3_s0me_f4st_r3l04ds}

Task 6

Link: http://58.65.172.131:5700/ - source_code

Part 1:

```
<!DOCTYPE html>
<html lang="en">
▼<head> == $0
    <meta charset="UTF-8">
    <title>JackSparrow</title>
    <!-- Part 1 - TISS{i_w-->
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/normalize/5.0.0/normalize.min.css">
    <script src="index.js"></script>
    <link rel="stylesheet" href="style.css">
  </head>
▼<body>
  ▼<div class="container"> flex
    ▶<figure class="full-height">…</figure>
    </div>
  </body>
</html>
```

html head

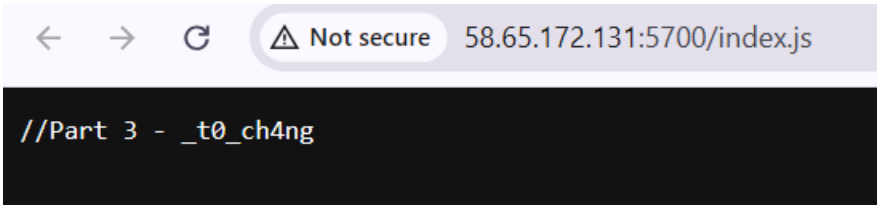
flag: TISS{i_w

Part 2:

```
            <div class="transparent">
                <!-- Part 2 - 0uld_1ov3 -->
            </div>
        </g>
        <g>
            <g transform="translate(497 984)">
                <circle cx="15" cy="15" r="15"/>
            </g>
        </g>
    </g>
```

flag: 0uld_1ov3

Part 3:



flag: _t0_ch4ng

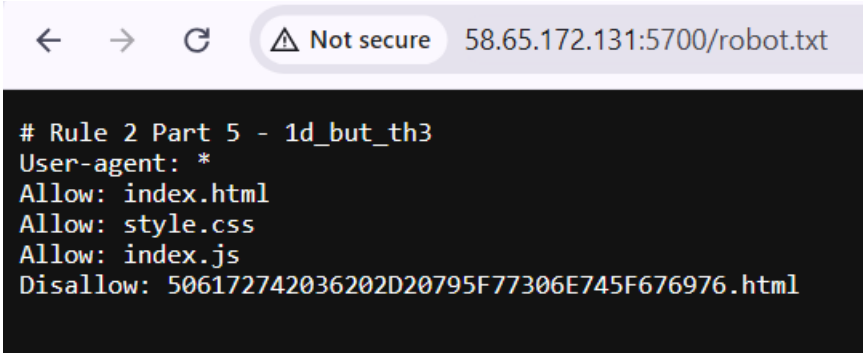
Part 4:

found in http://58.65.172.131:5700/style.css page:

```
.bubbles-large > g {
  -webkit-transform: translateY(2048px);
          transform: translateY(2048px);
          /*Part 4 - 3_th3_w0r*/
  opacity: 0;
  will-change: transform, opacity;
}
```

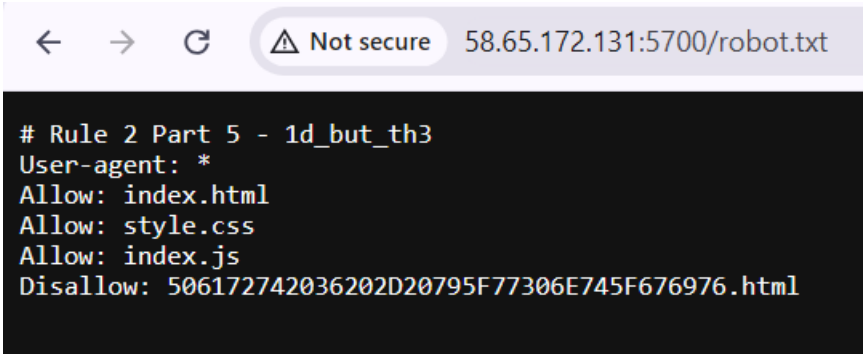
flag: 3_th3_w0r

Part 5:

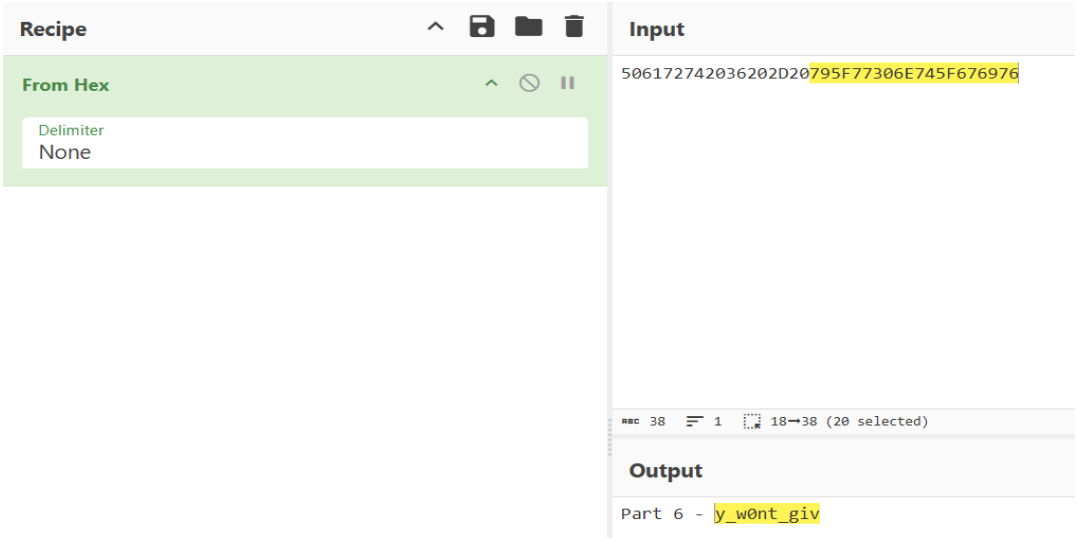


flag: 1d_but_th3

Part 6:

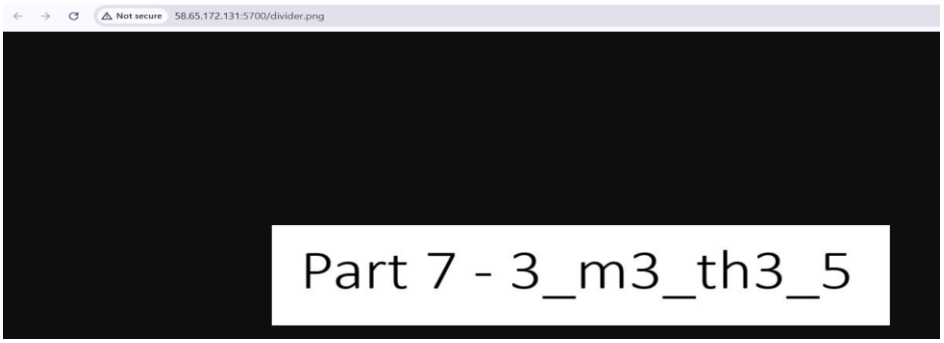


decoding 506172742036202D20795F77306E745F676976 gives:



flag: y_w0nt_giv

Part 7:

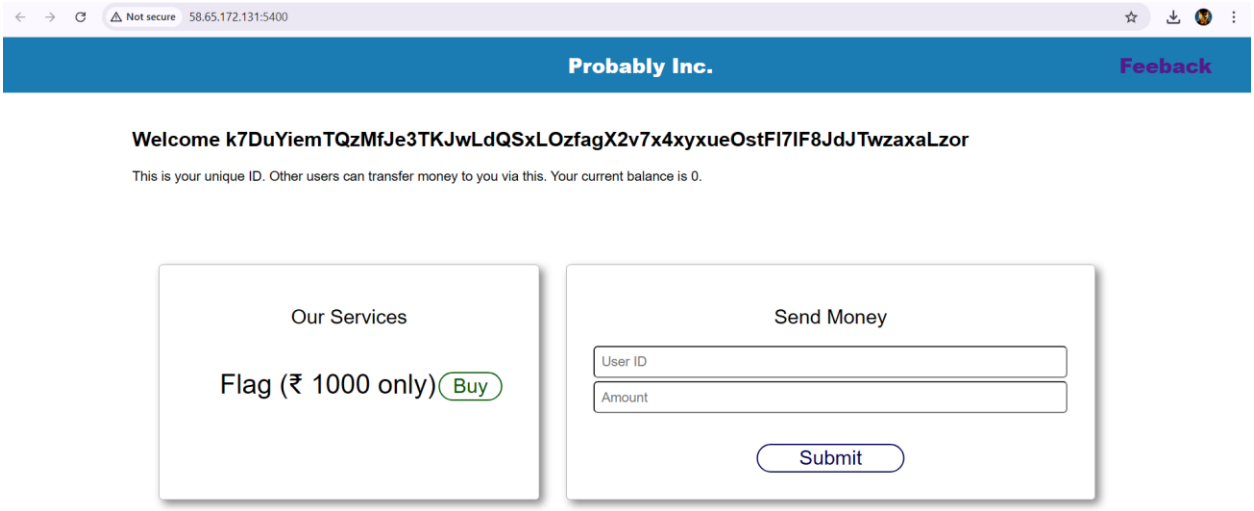


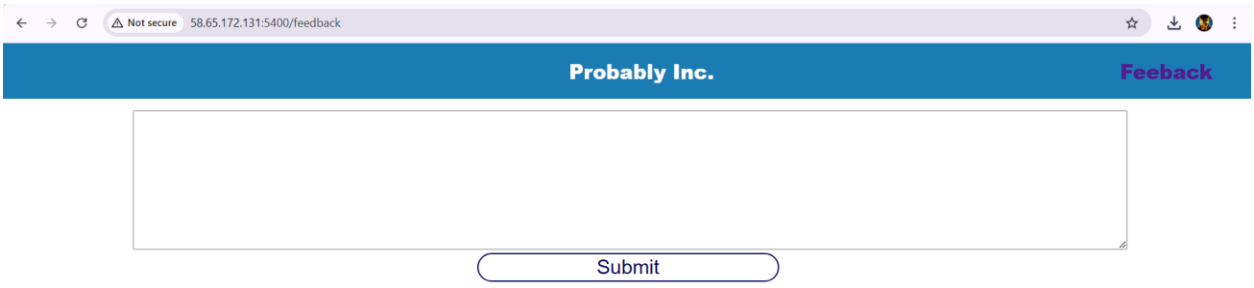
flag: 3_m3_th3_5

full flag: TISS{i_w0uld_1ov3_t0_ch4ng3_th3_w0r1d_but_th3 y_w0nt_giv3_m3_th3_5

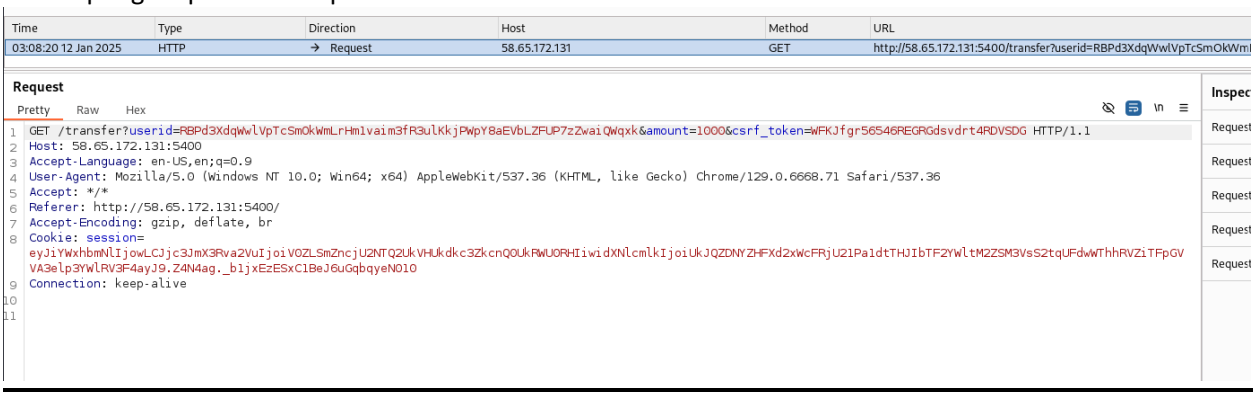
Task 4

Link: http://58.65.172.131:5400/ - csrf





Intercepting request on burpsuite:



decoding json token:

eyJiYWxhbmNlIjowLCJjc3JmX3Rva2VuIjoiV0Z
LSmZncjU2NTQ2UkVHUkdkc3ZkcnQ0UkRWU0RHIi
widXNlcmlkIjoiUkJQZDNYZHFXd2xWcFRjU21Pa
1dtTHJIbTF2YWltM2ZSM3VsS2tqUFdwWThhRVZi
TFpGVVA3elp3YWlRV3F4ayJ9.Z4N4ag._b1jxEz
ESxC1BeJ6uGqbqyeN010

{
"balance": 0,
"csrf_token": "WFKJfgr56546REGRGdsvdrt4RDVSDG",
"userid":
"RBPd3XdqWwlVpTcSmOkWmLrHm1vaim3fR3ulKkjPWpY8aEVbLZFUP7z
ZwaiQWqxk"
}

PAYLOAD: DATA