

ACM WEEK -3 CHALLENGES

-G.Yeshwanthi Reddy(AM.SC.U4CYS24020)

CHALL1:

Tools used : burp suit(proxy and repeater)

Process : Used burp suit to modify the http request (isadmin = 0 to isadmin = 1)

The screenshot shows the Burp Suite interface with the 'Intercept' tab selected. The timeline pane displays three requests:

- 08:20:04 \$ Au... HTTP → Request: GET http://localhost:3000/
- 08:20:09 \$ Au... HTTP → Request: GET http://localhost:3000/check.php
- 08:20:30 \$ Au... HTTP → Request: GET http://localhost:3000/check.php

The screenshot shows the Burp Suite interface with the 'Request' tab selected. The request pane displays the intercepted GET request to /check.php:

```
GET /check.php HTTP/1.1
Host: localhost:3000
sec-ch-ua: "Not)A;Brand";v="0", "Chromium";v="130"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-Secure: none
Sec-Fetch-Dest: document
Referer: http://localhost:3000/
```

Intercepted request to /check.php from proxy

Sent to repeater

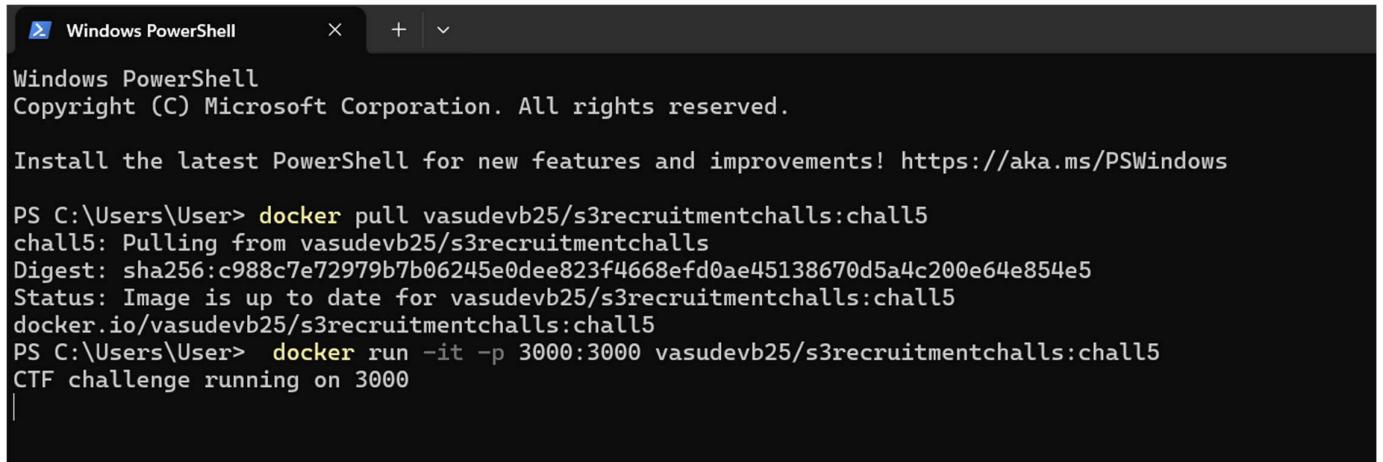
The screenshot shows the Burp Suite interface with the 'Response' tab selected. The response pane displays the modified response from the repeater:

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: text/html; charset=utf-8
Content-Length: 34
ETag: W/"22-b6PJMhCsabnWwPvRHN0YIWcns"
Date: Tue, 05 Aug 2025 02:50:51 GMT
Connection: keep-alive
Keep-Alive: timeout=5
<h4> ACM(cookie_monster_admin)</h4>
```

Changed the cookie isadmin = 0 to isadmin = 1 and sent the request...got the flag - **ACM{cookie_monster_admin}**

CHALL5:

Tools used : burpsuit

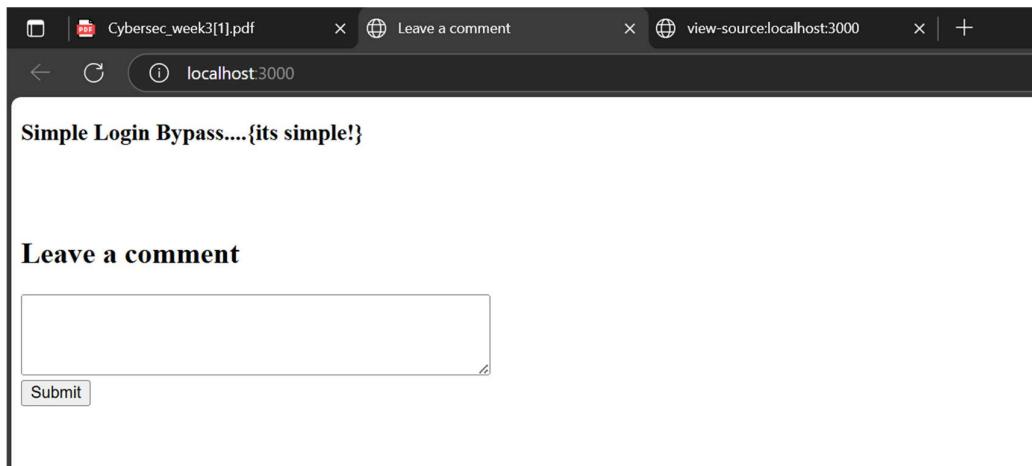


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

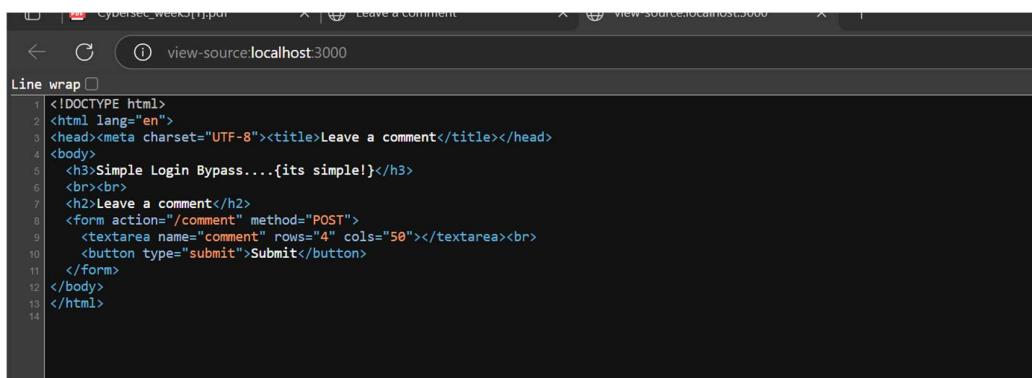
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\User> docker pull vasudevb25/s3recruitmentchalls:chall5
chall5: Pulling from vasudevb25/s3recruitmentchalls
Digest: sha256:c988c7e72979b7b06245e0dee823f4668efd0ae45138670d5a4c200e64e854e5
Status: Image is up to date for vasudevb25/s3recruitmentchalls:chall5
docker.io/vasudevb25/s3recruitmentchalls:chall5
PS C:\Users\User> docker run -it -p 3000:3000 vasudevb25/s3recruitmentchalls:chall5
CTF challenge running on 3000
```

Opened the browser:



Searched in page source for the flag:



```
<!DOCTYPE html>
<html lang="en">
<head><meta charset="UTF-8"><title>Leave a comment</title></head>
<body>
<h3>Simple Login Bypass....{its simple!}</h3>
<br><br>
<h2>Leave a comment</h2>
<form action="/comment" method="POST">
<textarea name="comment" rows="4" cols="50"></textarea><br>
<button type="submit">Submit</button>
</form>
</body>
</html>
```

AS it says simple bypass login ...I tried to run in burpsuit to find any flag

The screenshot shows the Burp Suite interface. In the Proxy tab, a POST request to `http://localhost:3000/comment` is selected. The request payload is "hlo". The browser window displays a page titled "Leave a comment" with the text "Simple Login Bypass....{its simple!}".

Intercepted the Get request and typed “hlo” in comment to get the POST request ..as we can see in the above pic we got the flag : **FLAG%7Bstolen_cookie_flag%7D**

CHALL3:

Tools used: page source

Process :

```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\User> docker pull vasudevb25/s3recruitmentchalls:chall3
chall3: Pulling from vasudevb25/s3recruitmentchalls
Digest: sha256:007fcd78ce5d51821c28f9ac7448bdb9574f9a4219162f13c221f48fd4c755f6
Status: Image is up to date for vasudevb25/s3recruitmentchalls:chall3
docker.io/vasudevb25/s3recruitmentchalls:chall3
PS C:\Users\User> docker run -it -p 3000:3000 vasudevb25/s3recruitmentchalls:chall3
CTF challenge running on 3000
```

Secure Customer Portal

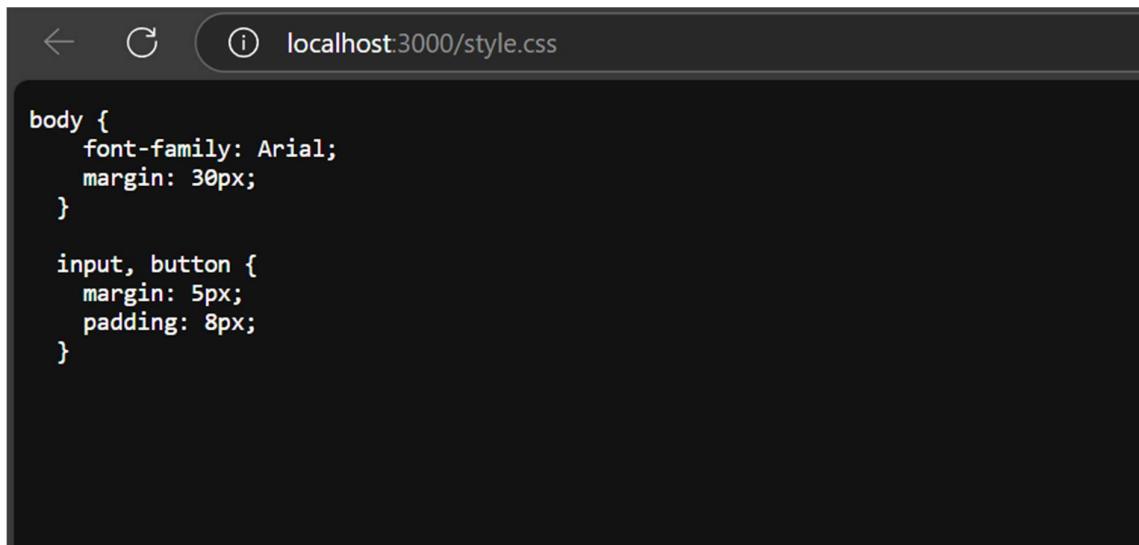
Only letters and numbers allowed for username and password.

Username
Password
<input type="button" value="Login"/>

Visited the page source to find any hidden scripts..etc related to flag

```
Line wrap □
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Secure Customer Portal</title>
6   <link rel="stylesheet" href="style.css">
7 </head>
8 <body>
9   <h1>Secure Customer Portal</h1>
10  <p>Only letters and numbers allowed for username and password.</p>
11
12 <form>
13   <input type="text" id="username" placeholder="Username" required><br>
14   <input type="password" id="password" placeholder="Password" required>
15   <button type="button" onclick="login()">Login</button>
16 </form>
17
18 <form hidden action="admin.php" method="POST" id="hiddenAdminForm">
19   <input type="text" name="hash" id="adminFormHash">
20 </form>
21
22 <p id="msg"></p>
23
24 <script src="secure.js"></script>
25 </body>
26 </html>
```

Here I found a javascript(secure.js) and css script(style.css) files..lets open it



A screenshot of a web browser window. The address bar shows "localhost:3000/style.css". The main content area displays the CSS code for "style.css".

```
body {
  font-family: Arial;
  margin: 30px;
}

input, button {
  margin: 5px;
  padding: 8px;
}
```

The above one is style.css script..

Nothing related to flag found from this...

```

function filter(input) {
    for (let i = 0; i < input.length; i++) {
        let cc = input.charCodeAt(i);
        if (!(cc >= 48 && cc <= 57) || (cc >= 65 && cc <= 90) || (cc >= 97 && cc <= 122))) {
            return false;
        }
    }
    return true;
}

function checkPassword(username, password) {
    return username === "admin" && password === "strongpassword";
}

function login() {
    const user = document.getElementById("username").value;
    const pass = document.getElementById("password").value;

    if (!filter(user) || !filter(pass)) {
        document.getElementById("msg").innerText = "Illegal character in username or password.";
        return;
    }

    if (checkPassword(user, pass)) {
        document.getElementById("msg").innerText = "Login Successful!";
        document.getElementById("adminFormHash").value = "2196812e91c29df34f5e217cf639881";
        document.getElementById("hiddenAdminForm").submit();
    } else {
        document.getElementById("msg").innerText = "Login Failed!";
    }
}

```

The above one is secure.js script ...here we can see username = admin , password = strongpassword

Entered the login credentials :

ACM{client_side_authentication_is_bad}



Got the flag : **ACM{client_side_authentication_is_bad}**

CHALL2:

Tools used : page source,base64 decoder

Process :

```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\User> docker pull vasudevb25/s3recruitmentchalls:chall2
chall2: Pulling from vasudevb25/s3recruitmentchalls
Digest: sha256:644a0235de00a9eb6445a173461dd29eceadb5c194fd6c78b80f0ff974a00fc
Status: Image is up to date for vasudevb25/s3recruitmentchalls:chall2
docker.io/vasudevb25/s3recruitmentchalls:chall2
PS C:\Users\User> docker run -it -p 3000:3000 vasudevb25/s3recruitmentchalls:chall2
CTF challenge running on 3000
|
```

The screenshot shows a browser window with the URL `localhost:3000` in the address bar. The page content includes:

- A navigation bar with links: Home, About, Contact.
- A heading: **Entha Johnsa kalille? kallumakayile?**
- A large image of a man with a mustache.
- Text: "Flag njan olipichu" and "Haaaaaaaaaaaaaaaaaaaaaaaaaaaaaa".
- A section titled **ADICHU KERI VAAAAA**.
- A small image of a man with a mustache, with the text "Don't give up!" below it.

Opened the browser and headed to `localhost:3000`

Then got into **about** page and viewed its page source

```
Line wrap □
1 <div style="background-color: aqua; min-height: 100vh; position: relative;">
2   <section class="about" notify_true="Um14aFozdHBibk53WldOMFgyWnNZV2RmWTJoaGJtZGxmUT09">
3     <h1 style="text-align: center; font-size: 50px;">U HAVE BEEN MOGGED BY</h1>
4     
5     <footer>
6       <h2>HAHAHAHAHA!!!!</h2>
7       <h3>Nah just kidding!</h3>
8     </footer>
9   </section>
10 </div>
11
```

From the page source I found a base64 encoded script

“Um14aFozdHBibk53WldOMFgyWnNZV2RmWTJoaGJtZGxmUT09”

Decoded it 2 times using cyberchef

The screenshot shows two instances of CyberChef's "Input" field containing the string "Um14aFozdHBibk53WldOMFgyWnNZV2RmWTJoaGJtZGxmUT09". Below each input is an "Output" field. The first output shows the string "RmxhZ3tpbnNwZWN0X2ZsYWdfY2hhbmdlfQ==". The second output shows the string "Flag{inspect_flag_change}".

Flag obtained is **Flag{inspect_flag_change}**

CHALL4:

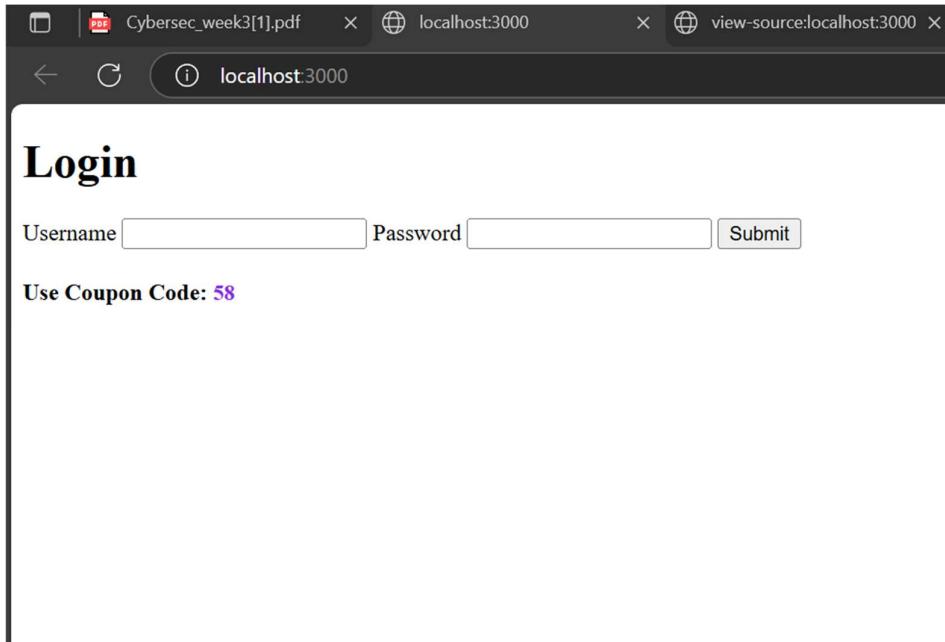
Tools used : page source,base58 decoder

Process :

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\User> docker pull vasudevb25/s3recruitmentchalls:chall4
chall4: Pulling from vasudevb25/s3recruitmentchalls
Digest: sha256:ac0b22a79ae0d88c69a11a3b7ba8e774630decaed17994d13bd1635a5a8f80f
Status: Image is up to date for vasudevb25/s3recruitmentchalls:chall4
docker.io/vasudevb25/s3recruitmentchalls:chall4
PS C:\Users\User> docker run -it -p 3000:3000 vasudevb25/s3recruitmentchalls:chall4
CTF challenge running on 3000
```



Viewed page source and found a javascript(index.js)

```
Line wrap □
1 <!doctype html>
2 <html>
3   <head>
4     <script src="index.js"></script>
5   </head>
6   <body>
7     <div>
8       <h1>Login</h1>
9       <form method="POST">
10         <label for="username">Username</label>
11         <input name="username" type="text"/>
12         <label for="password">Password</label>
13         <input name="password" type="password"/>
14         <input type="submit" value="Submit"/>
15     </form>
16   </div>
17   <h4>Use Coupon Code: <b style="color: blueviolet; font-size: 15px;">58 </b></h4>
18 </body>
19 </html>
```

Viewed index.js:

```
(async()=>{function delay(ms){return new Promise(res=>setTimeout(res,ms));}const handler={eventType:"load",listener:(cb)=>window.addEventListener("load",cb)};async function
onReady(){await new Promise(resolve=>(handler.listener(()=>(delay(50).then(resolve);))));}function encodeValue(input){const base64=bttoa(input);const
stripped=base64.replace(/-/g,"");const noise=stripped.split("").map(char=>char).join("");return noise;}function revealFlag(encoded){const
decoded=atob(encoded);console.log("Decrypting...");return decoded;await onReady();const selectors={u:"input[name=username]",p:"input[name=password]"};const keySet=
["BzJT2Y5","7Qheif3MWxVVHXpM59BfcXpikLvR4DkH2"];function auditLog(data){console.debug("Captured Input ->",data);}document.querySelector("form").addEventListener("submit",
(event)=>(event.preventDefault());const inputs={};for(const key in selectors){const
val=document.querySelector(selectors[key]).value;inputs[key]=encodeValue(val);}auditLog(inputs);const checkUser=()=>inputs.u==keySet[0];const checkPass=
()=>inputs.p==keySet[1];return!checkUser()?alert("Incorrect Username"):!checkPass()?alert("Incorrect Password"):void alert('Correct Password! Your flag is
${revealFlag(inputs.p)}');});})();}
```

From this we can see a base58 encoded script

"BzJT2Y5","7Qheif3MWxVVHXpM59BfcXpikLvR4DkH2"

Now decode it with cyberchef

The image shows two terminal windows side-by-side. Both windows have tabs labeled 'Input' and 'Output'. The top window's 'Input' tab contains the text 'BzJT2Y5|'. The bottom window's 'Input' tab contains the text '7Qheif3MwxVVHXpM59BfcXpikLvR4Dkh2|'. Both windows have a status bar at the bottom showing 'Raw Bytes' and 'LF'. The 'Output' tabs of both windows show the command 'FLAG{base64_is_outdated}'.

```
BzJT2Y5|
nsc 7  ⏎ 1
Tr Raw Bytes ⏵ LF
Output
admin|
```



```
7Qheif3MwxVVHXpM59BfcXpikLvR4Dkh2|
nsc 33  ⏎ 1
Tr Raw Bytes ⏵ LF
Output
FLAG{base64_is_outdated}|
```

By decoding base58 script we got the flag as **FLAG{base64_is_outdated}**