

HTTP协议

1. 请求方式
2. 302跳转
3. Cookie
4. 基础认证
5. 响应包源代码

HTTP协议

1. 请求方式

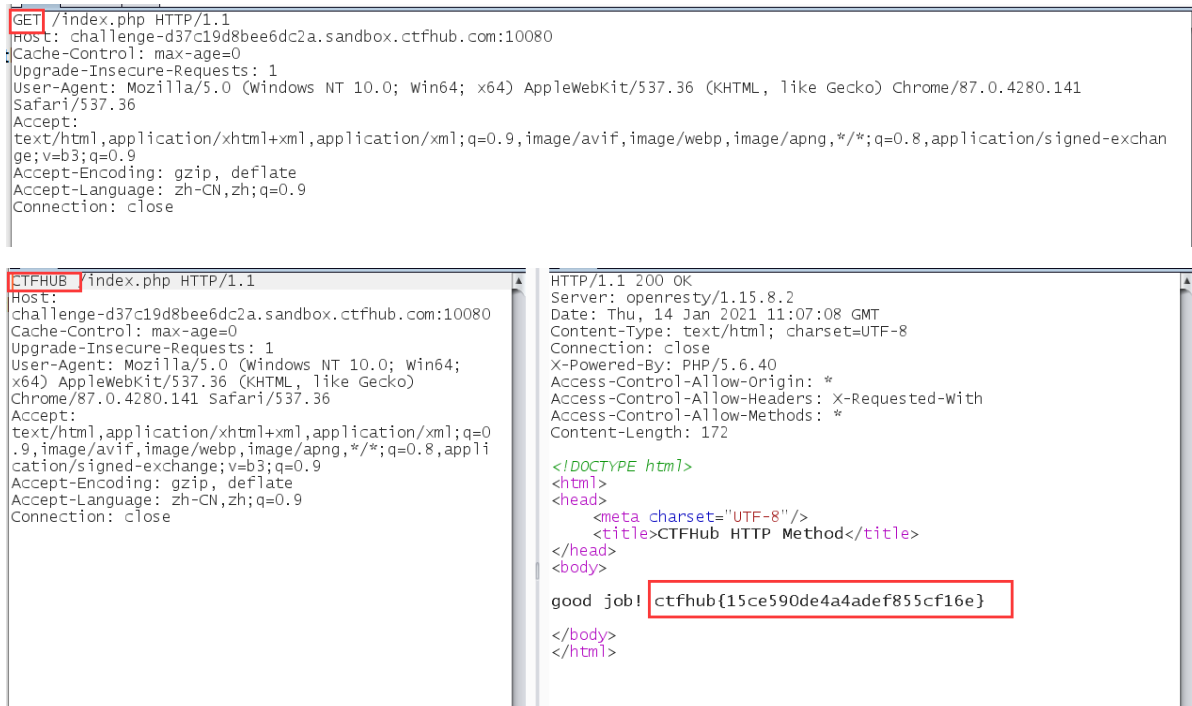
HTTP Method is GET

Use **CTF**B** Method, I will give you flag.

Hint: If you got 「HTTP Method Not Allowed」 Error, you should request index.php.

他说使用CTF**B请求方式，估计是CTFHUB。

直接将 GET 改成 CTFHUB 估计就可以了。



```
GET /index.php HTTP/1.1
Host: challenge-d37c19d8bee6dc2a.sandbox.ctfhub.com:10080
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.141 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=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close

CTFHUB /index.php HTTP/1.1
Host: challenge-d37c19d8bee6dc2a.sandbox.ctfhub.com:10080
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.141 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=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close

HTTP/1.1 200 OK
Server: openresty/1.15.8.2
Date: Thu, 14 Jan 2021 11:07:08 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/5.6.40
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: X-Requested-With
Access-Control-Allow-Methods: *
Content-Length: 172

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8"/>
  <title>CTFHub HTTP Method</title>
</head>
<body>
  good job! ctfhub{15ce590de4a4adef855cf16e}
</body>
</html>
```

果然成功，得到了flag。

2. 302跳转

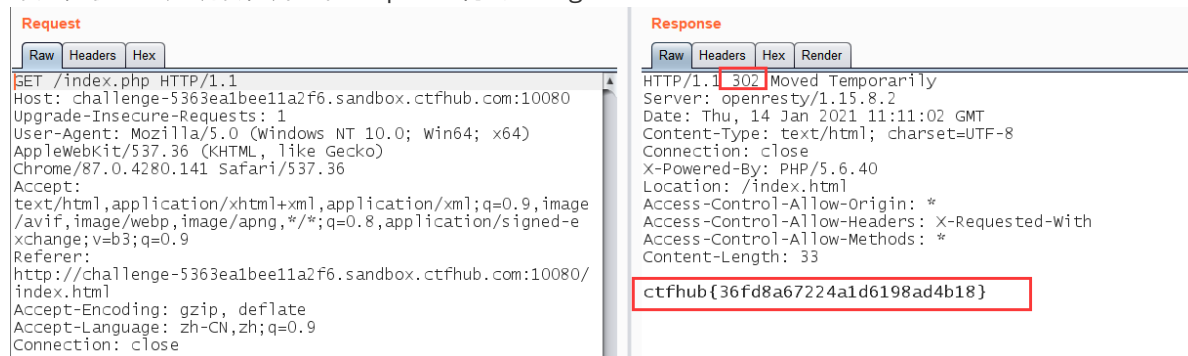
进主页之后，长这样，题目又是302跳转，估计是跳转的中间存在flag。

No Flag here!

[Give me Flag](#)

尝试用burp抓包。

果然，状态码是跳转，并且在response得到了flag。

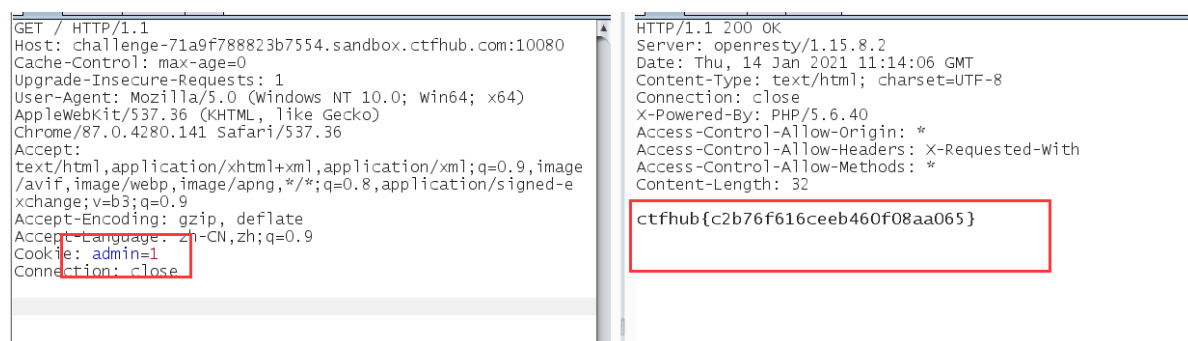


3. Cookie

hello guest. only admin can get flag.

提示是，只有admin可以拿到flag，不妨将cookie改一下。将admin的值从0改成1。

成功得到cookie。



4. 基础认证

这题还有个附件，附件里是一些常用的弱密码。估计就是直接爆破。

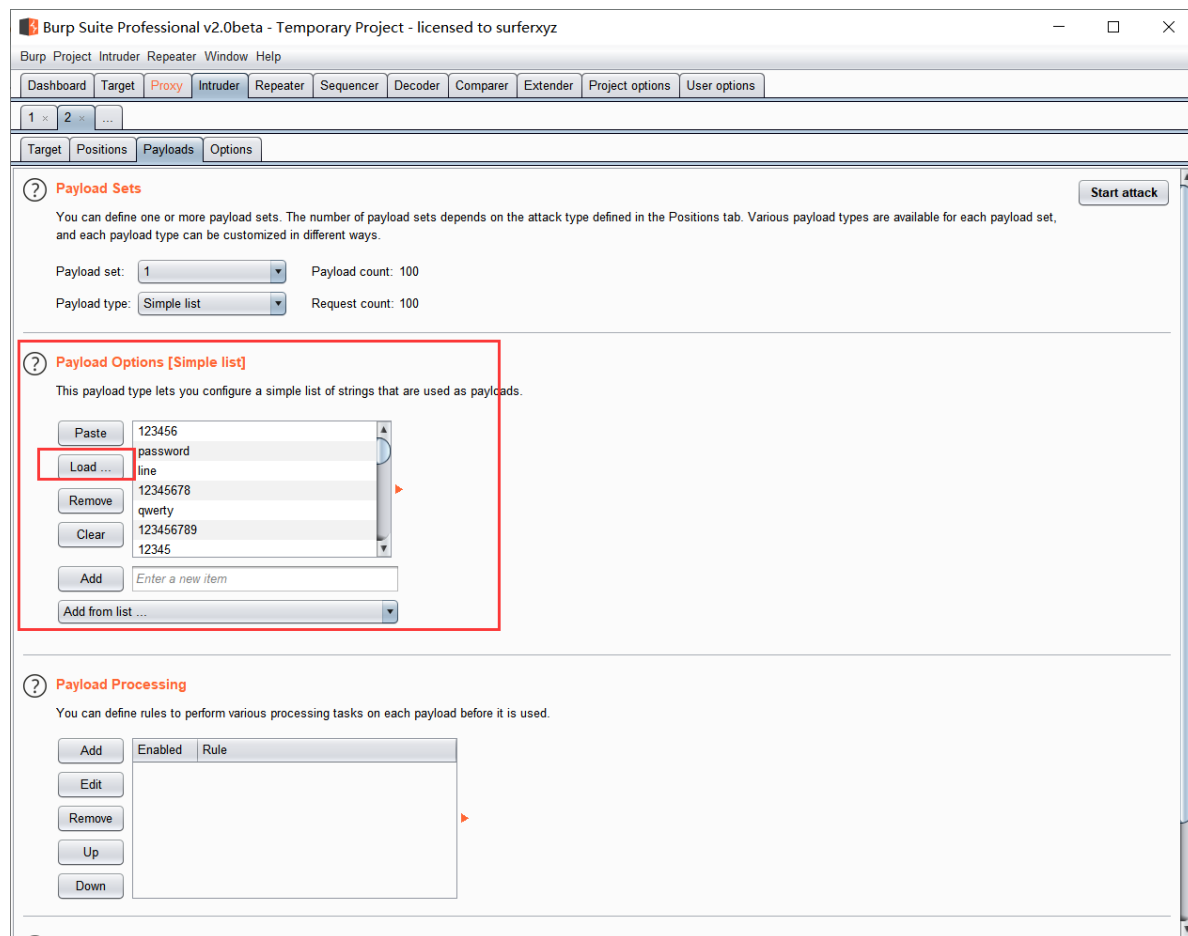
输入密码后，用burp抓包，虽然没看到一般的密码，但是看到了一串base64加密的东西，猜测那个就是我输入的密码。



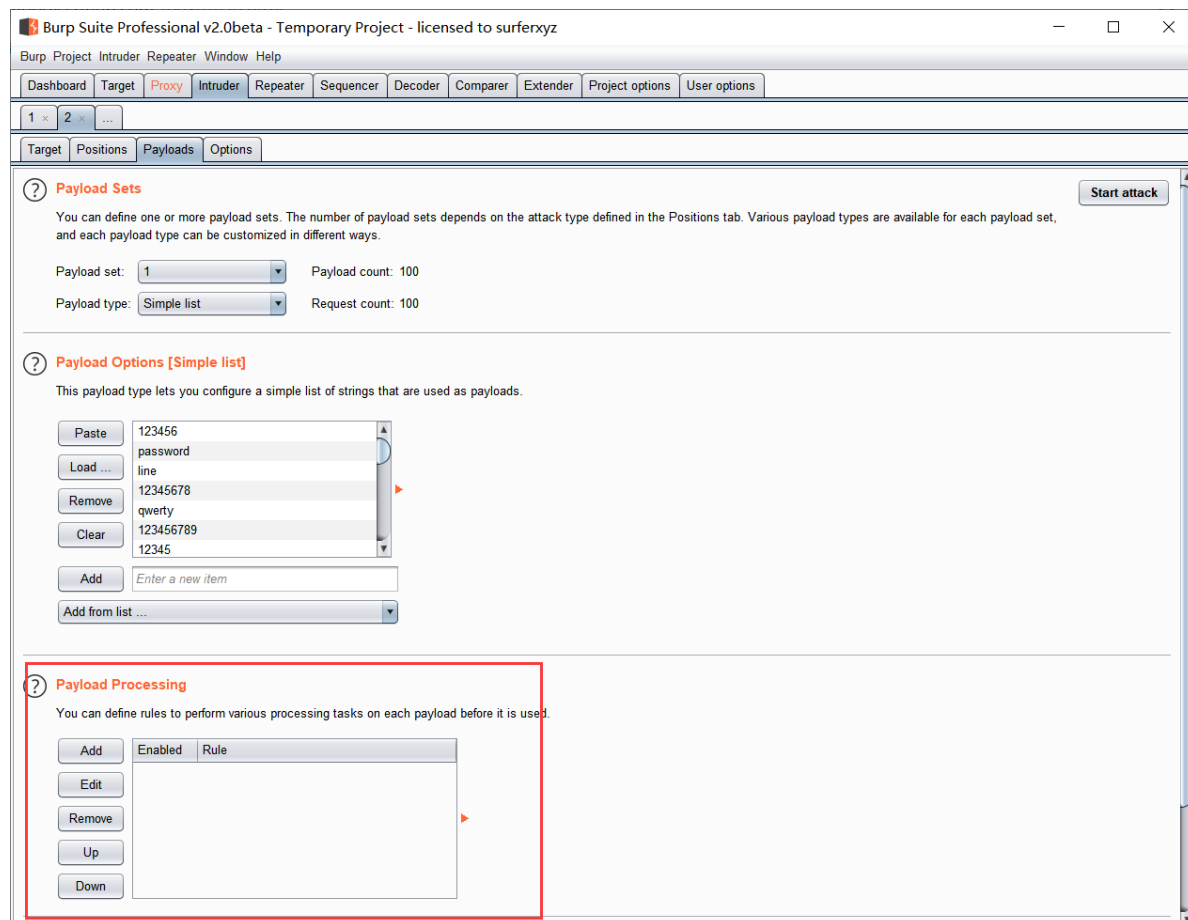
果然，解密之后发现这就是我输入的账号和密码。

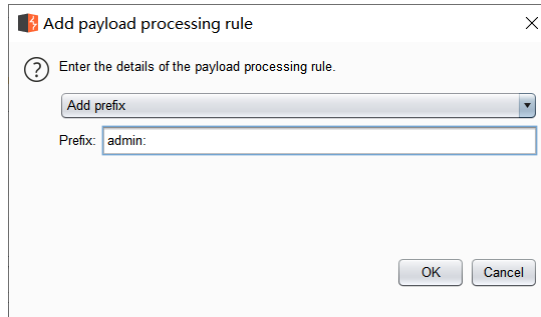
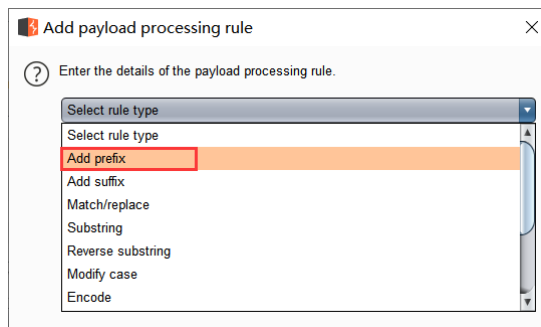
但由于附件只给了密码，所以用户名还是直接用常用的admin。进行爆破。

首先将密码load进去。

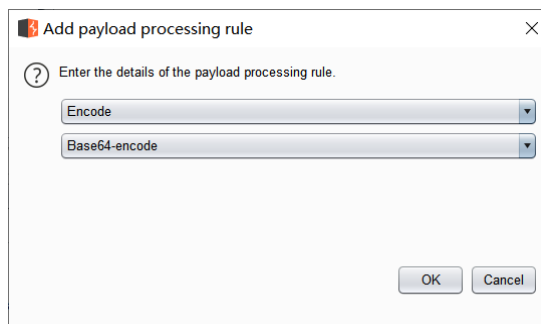


然后在密码前面加上 admin: 。

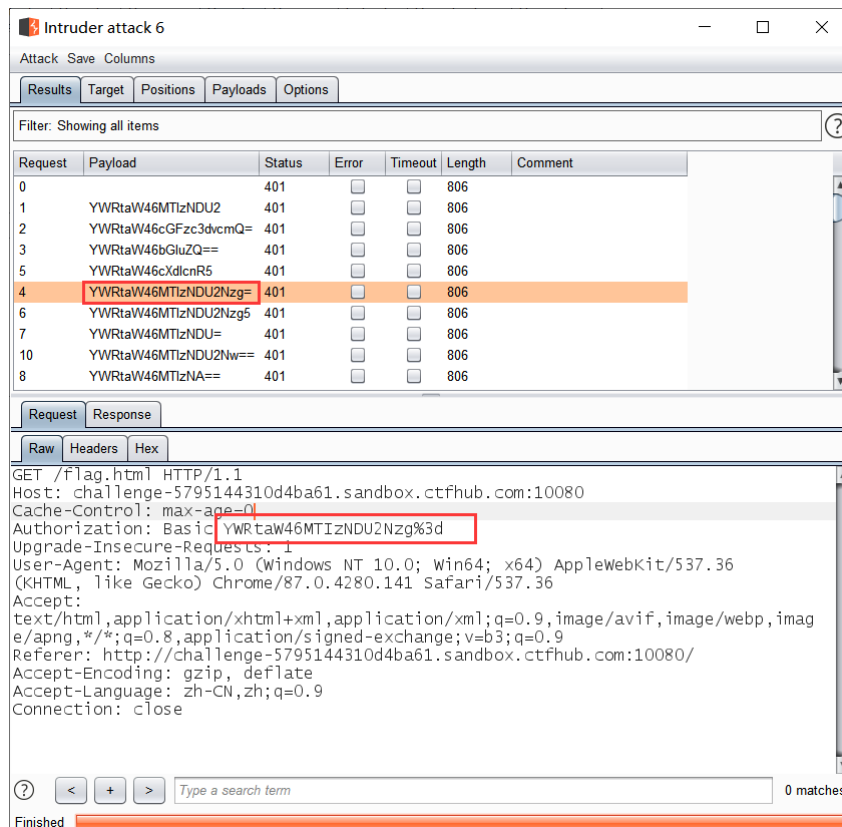




再将整个进行base64-encode。



然后爆破，却发现全部都不行。



发现在编码的时候，把所有的=都给替换成了%3d(URL编码)。

? Payload Encoding

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

☒ URL-encode these characters: `./!<>?+&*,:;"'[]{}^``

最后将这个勾给去掉，不让他进行url-encode。（或者把里面的=给删了）

这下就发现了一个可以的。

The screenshot shows the 'Intruder attack 8' window in Burp Suite. The 'Results' tab is active, displaying a table of requests. The first request (index 61) is selected, and its details are shown in the 'Request' tab below. The payload 'YWRtaW46a2xhc3Rlcg==' is highlighted in the table and in the 'Authorization' header of the request details.

Request	Payload	Status	Error	Timeout	Length	Comment
61	YWRtaW46a2xhc3Rlcg==	200			378	
0		401			806	
1	YWRtaW46MTIzNDU2	401			806	
2	YWRtaW46cGFzc3dvcmQ=	401			806	
3	YWRtaW46bGlzQ==	401			806	
4	YWRtaW46MTIzNDU2Nzg=	401			806	
5	YWRtaW46cXdlcnR5	401			806	
6	YWRtaW46MTIzNDU2Nzg5	401			806	
7	YWRtaW46MTIzNDU=	401			806	
8	YWRtaW46MTIzNA==	401			806	

Request Details:

```
GET /flag.html HTTP/1.1
Host: challenge-5795144310d4ba61.sandbox.ctfhub.com:10080
Cache-Control: max-age=0
Authorization: Basic YWRtaW46a2xhc3Rlcg==
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.141 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=0.9
Referer: http://challenge-5795144310d4ba61.sandbox.ctfhub.com:10080/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close
```

这样就破解出了密码，并且得到了flag。

The screenshot shows the Burp Suite interface. The 'Intruder attack 8' window is open, showing the same list of requests as before. The 'Proxy' tab is also visible, showing a list of intercepted requests. The first request is highlighted, and its details are shown in the 'Request' tab. The payload 'admin:klaster' is highlighted in the 'Request' tab, and the decoded payload 'ctfhub{4ab660b0caa173cb488f6043}' is shown in the 'Response' tab.

Proxy Tab:

YWRtaW46a2xhc3Rlcg==

admin:klaster

Intruder attack 8:

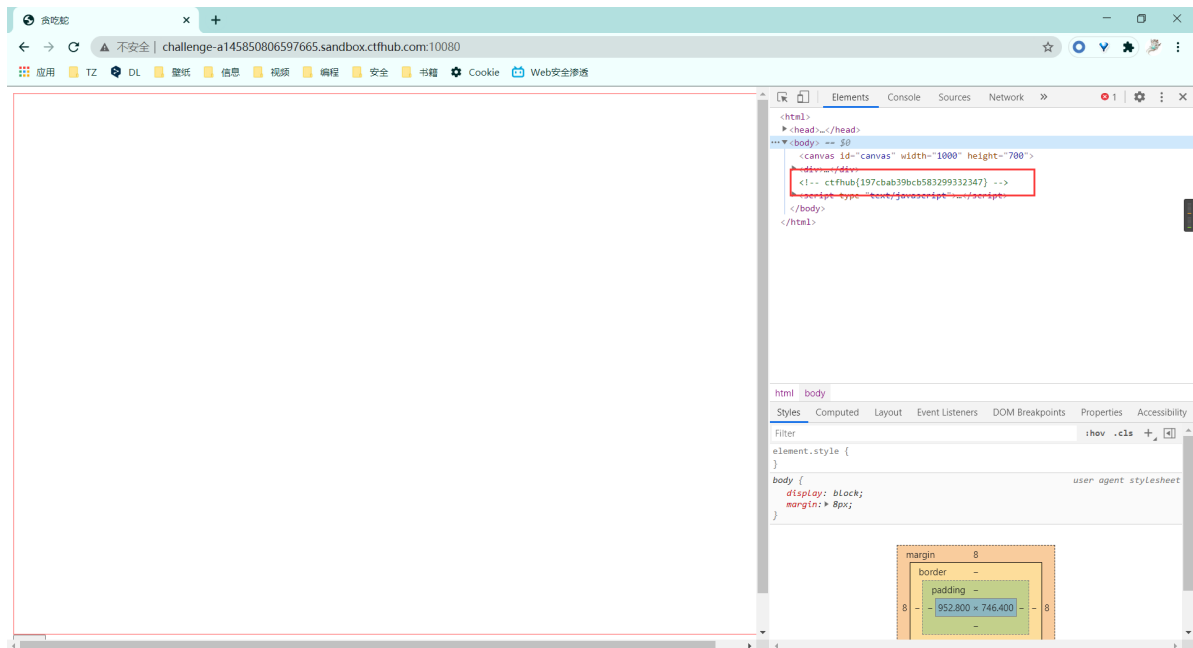
Request	Payload	Status	Error	Timeout	Length	Comment
61	YWRtaW46a2xhc3Rlcg==	200			378	
0		401			806	
1	YWRtaW46MTIzNDU2	401			806	
2	YWRtaW46cGFzc3dvcmQ=	401			806	
3	YWRtaW46bGlzQ==	401			806	
4	YWRtaW46MTIzNDU2Nzg=	401			806	
5	YWRtaW46cXdlcnR5	401			806	
6	YWRtaW46MTIzNDU2Nzg5	401			806	
7	YWRtaW46MTIzNDU=	401			806	
8	YWRtaW46MTIzNA==	401			806	

Request Details:

```
HTTP/1.1 200 OK
Server: openresty/1.15.8.2
Date: Thu, 14 Jan 2021 11:35:19 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Last-Modified: Thu, 14 Jan 2021 11:14:44 GMT
ETag: W/"600027a4-21"
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: X-Requested-With
Access-Control-Allow-Methods: *
Content-Length: 33

ctfhub{4ab660b0caa173cb488f6043}
```

5. 响应包源代码



按F12直接查看源代码，得到了flag。