

CTFHub Http协议

原创

日月ton光 于 2020-03-08 15:02:22 发布 2397 收藏 13

分类专栏: [CTF](#) 文章标签: [http协议](#) [http请求方式](#) [cookie欺骗](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/sinat_36711025/article/details/104733085

版权



[CTF 专栏收录该内容](#)

3 篇文章 1 订阅

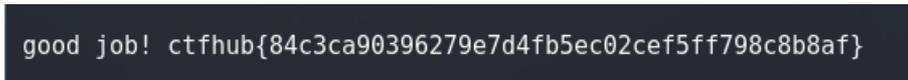
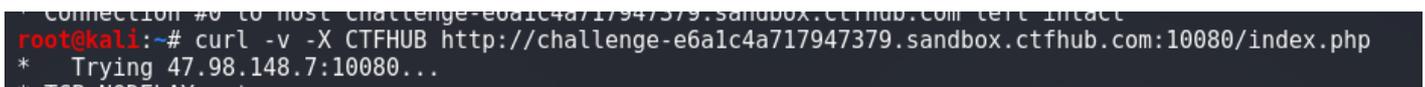
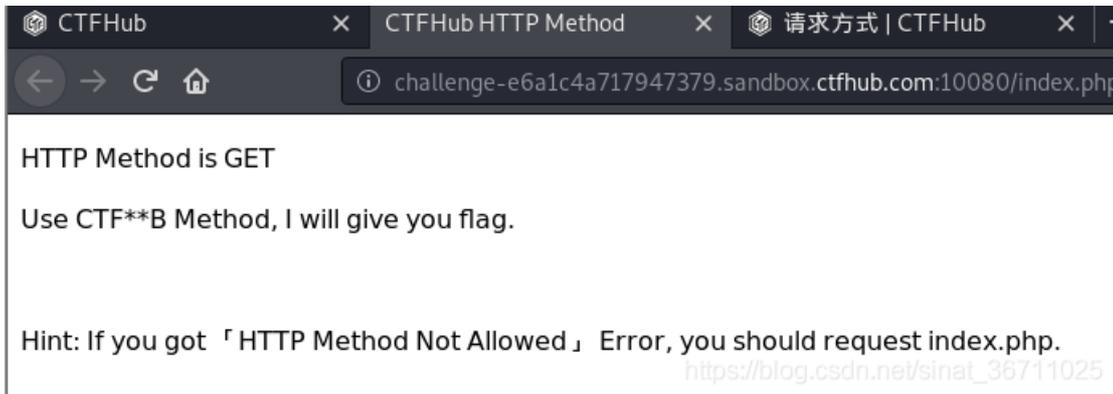
订阅专栏

1、请求方式

由题目可知, 考察的是HTTP请求方法, 在Http/1.1协议中定义的八种方法为GET, POST、HEAD、OPTIONS, PUT, DELETE, TRACE 和 CONNECT 方法。

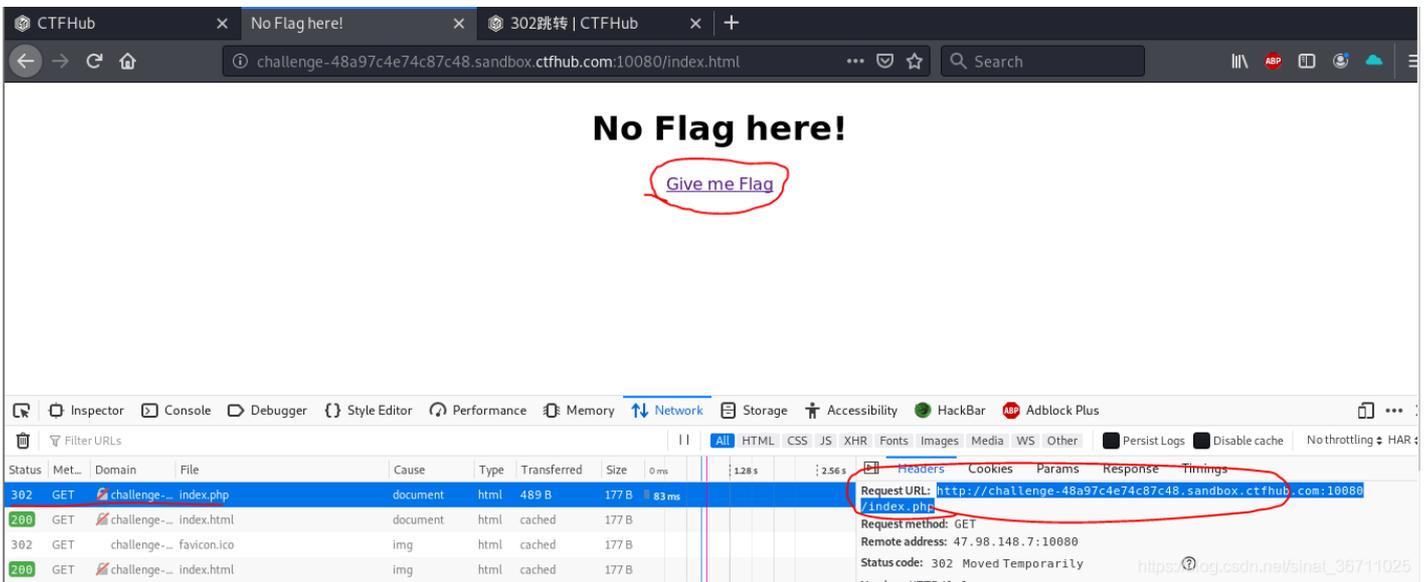


进入题目后, 发现提示, HTTP Method 是可以自定义的, 并且区分大小写, 直接用 CTFHUB 方法请求 index.php 即可拿到 flag。



2、302跳转

302是Http协议的临时重定向状态码。进入页面后发现没有任何消息, 按f12进入控制台, 点击Give me Flag按钮, 发现页面没有任何变化, 但其实页面已经发生了跳转, 用curl访问新的跳转页面即可得到flag值。



```

root@kali:~# curl -v http://challenge-48a97c4e74c87c48.sandbox.ctfhub.com:10080/index.php
* Trying 47.98.148.7:10080

```

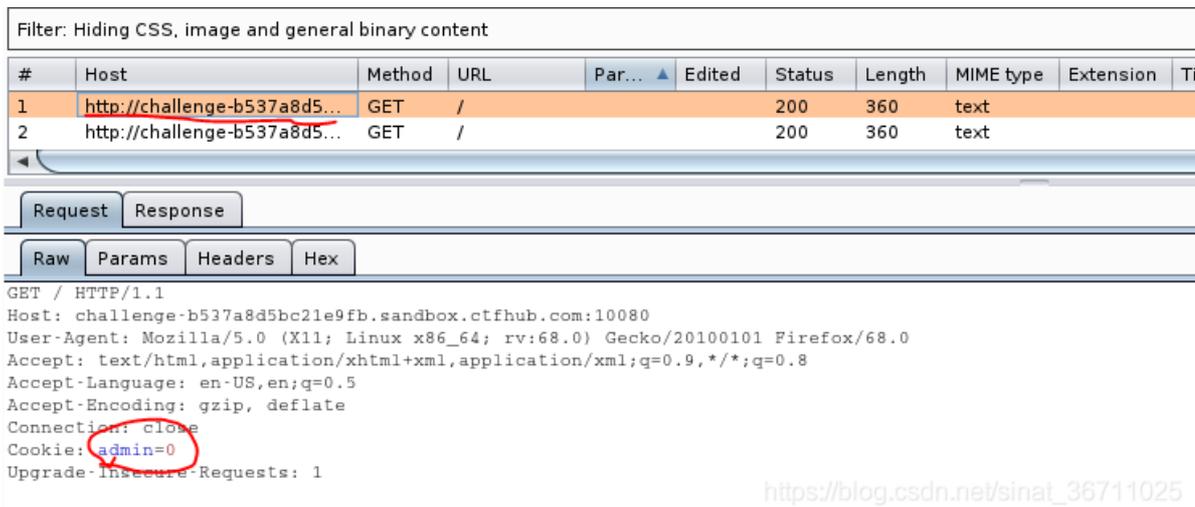
```

ctfhub{a39831bab3494820a0320b3ea1ae526cfalb4757}
* Connection #0 to host challenge-48a97c4e74c87c48.sandbox.ctfhub.com left intact

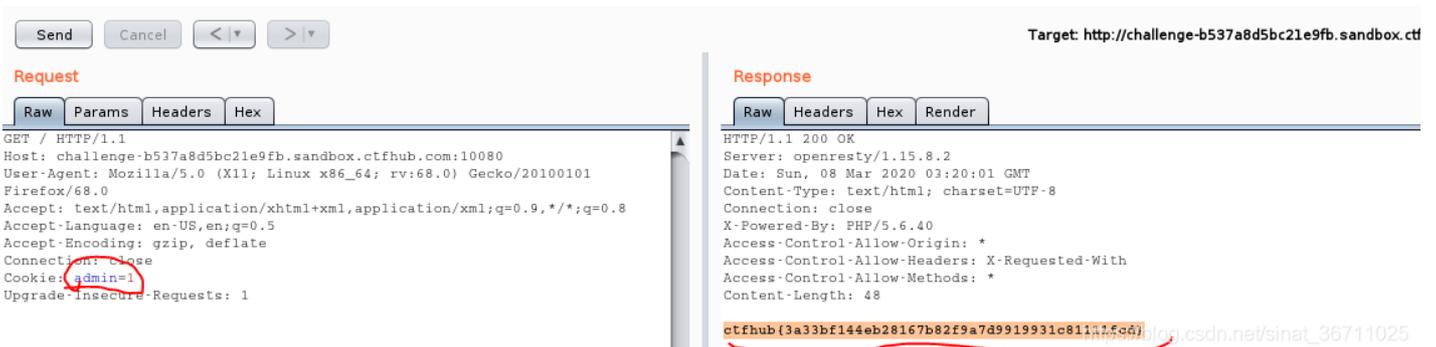
```

3、cookie

访问题目页面发现给的提示是“hello guest, only admin can get flag”，我们挂上 BurpSuite 之后重新访问题目页面，在cookie发现admin=0字段，猜测服务器可能是根据此字段来判断浏览者身份。



我们把这个请求放入“repeater”中修改Cookie: admin=0为Cookie: admin=1后发送，发现在响应页面里包含Flag。



4、基本认证

在HTTP中，基本认证（英语：Basic access authentication）是允许http用户代理（如：网页浏览器）在请求时，提供用户和密码的一种方式。详情请查看 <https://zh.wikipedia.org/wiki/HTTP基本认证>。

原理 [\[编辑\]](#)

文字过程 [\[编辑\]](#)

这一个典型的HTTP客户端和HTTP服务器的对话，服务器安装在同一台计算机上（localhost），包含以下步骤：

1. 客户端请求一个需要身份认证的页面，但是没有提供用户名和密码。这通常是用户在地址栏输入一个URL，或是打开了一个指向该页面的链接。
2. 服务端响应一个401应答码^[4]，并提供一个认证域（英语：Access Authentication）^[5]，头部字段为：WWW-Authenticate，该字段为要求客户端提供适配的资源。^[6] WWW-Authenticate: Basic realm="Secure Area" 该例子，Basic 为验证的模式，realm="Secure Area" 为保护域，用于与其他请求URI作区别。
3. 接到应答后，客户端显示该认证域给用户并提示输入用户名和密码。此时用户可以选择确定或取消。
4. 用户输入了用户名和密码后，客户端软件将对其进行处理，并在原先的请求上增加认证消息头（英语：Authorization）然后重新发送再次尝试。过程如下：

1. 将用户名和密码拼接为 用户: 密码 形式的字符串。
2. 如果服务器 WWW-Authenticate 字段有指定编码，则将字符串编译成对应的编码（如：UTF-8）。
3. 将字符串编码为base64。

4. 拼接 Basic，放入 Authorization 头字段，就像这样：Authorization Basic 字符串。示例：用户名：Aladdin，密码：OpenSesame，拼接后为 Aladdin:OpenSesame，编码后 QWxhZGRpbjPcGVuU2VzYW11，在HTTP头部里会是这样：Authorization: Basic QWxhZGRpbjPcGVuU2VzYW11。Base64编码并非加密算法，其无法保证安全与隐私，仅用于将用户名和密码中的不兼容的字符转换为均与HTTP协议兼容的字符集。

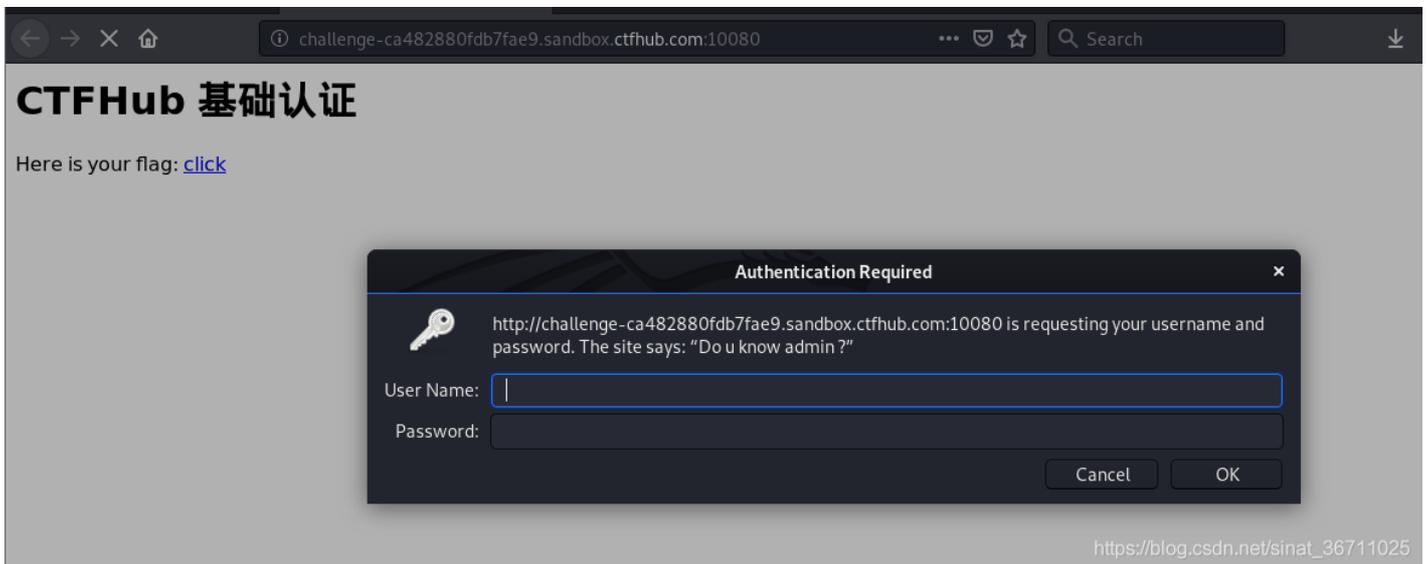
5. 在本例中，服务器接受了该认证屏幕并返回了页面。如果用户凭据非法或无效，服务器可能再次返回401应答码，客户端可以再次提示用户输入密码。

注意:客户端有可能不需要用户交互，在第一次请求中就发送认证消息头。



https://blog.csdn.net/sinat_CTFHub

进入题目页面，发现点击click会出现认证页面，



https://blog.csdn.net/sinat_36711025

挂上 BurpSuite 的代理，随便输个账号密码（比如：账号aaa 密码 bbb）访问，查看 HTTP 响应报文：

Intercept HTTP history WebSockets history Options

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	T
44	http://challenge-2fb9bdbe29db055e.sandbox.ctfhub.com	GET	/flag.html			404	1468	HTML	html	C
43	http://detectportal.firefox.c...	GET	/success.txt			200	270	text	txt	

Request Response

Raw Headers Hex

```

GET /flag.html HTTP/1.1
Host: challenge-2fb9bdbe29db055e.sandbox.ctfhub.com:10080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://challenge-2fb9bdbe29db055e.sandbox.ctfhub.com:10080/
Connection: close
Upgrade-Insecure-Requests: 1
Authorization: Basic YWFhOmJiYg==

```

https://blog.csdn.net/sinat_36711025

Request Response

Raw Headers Hex HTML Render

```

HTTP/1.1 401 Unauthorized
Server: openresty/1.15.8.2
Date: Sun, 08 Mar 2020 03:50:02 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 179
Connection: close
WWW-Authenticate: Basic realm="Do u know admin ?"

```

得到提示 do u konw admin ?，于是猜测账号是 admin，那么接下来就只需要爆破密码了。注意看到 HTTP 请求头部的 Authorization 字段，后面的 YWFhOmJiYg== 用 base64 解码后是 aaa:bbb，也就是我们之前输入的账号：密码。

使用 BurpSuite 进行基础认证爆破

(1) 将报文发送到 Intruder, 将 Basic 后面 base64 部分添加为 payload position

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type:

```

GET /flag.html HTTP/1.1
Host: challenge-2fb9bdbe29db055e.sandbox.ctfhub.com:10080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://challenge-2fb9bdbe29db055e.sandbox.ctfhub.com:10080/
Connection: close
Upgrade-Insecure-Requests: 1
Authorization: Basic YWFhOmJiYg==

```

Add \$
Clear \$
Auto \$
Refresh

https://blog.csdn.net/sinat_36711025

(2) 在 Payloads 选项卡下，选择 Payload Type 为 SimpleList，然后在 Payload Options 中点击 load 加载密码字典

② Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Buttons: Paste, Load ..., Remove, Clear, Add, Add from list ... [Pro version only]

URL: https://blog.csdn.net/sinat_36711025

(3) Payload Processing -> Add-> Add Prefix (添加前缀) -> 输入 admin:

Payload Processing -> Add-> Encode (添加一个编码方式) -> 选择 Base64 Encode

② Payload Processing

You can define rules to perform various processing tasks on each payload before it i

Enabled	Rule
<input checked="" type="checkbox"/>	Add Prefix: admin:
<input checked="" type="checkbox"/>	Base64-encode

Buttons: Add, Edit, Remove, Up, Down

URL: https://blog.csdn.net/sinat_36711025

(4) Payload Encode 取消勾选的 URL-encode, 不然你会看到base64之后的=会被转成 %3d, 你就算爆破到天荒地老也不会出来

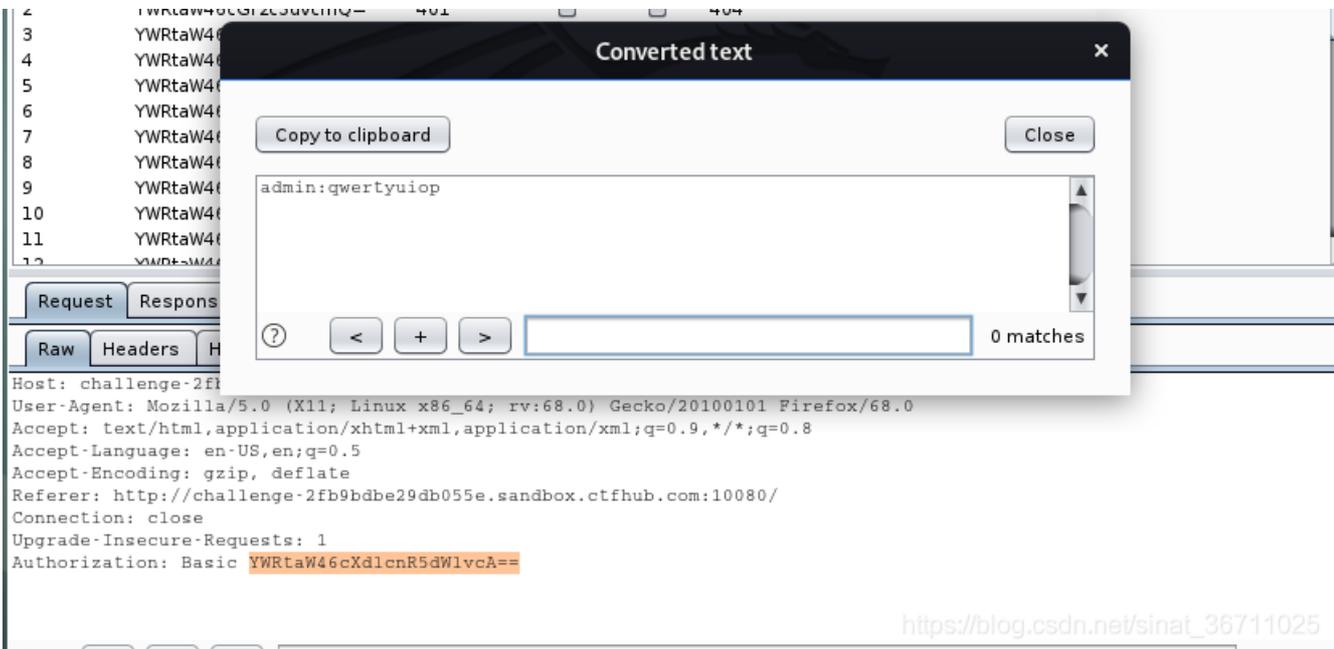
② Payload Encoding

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

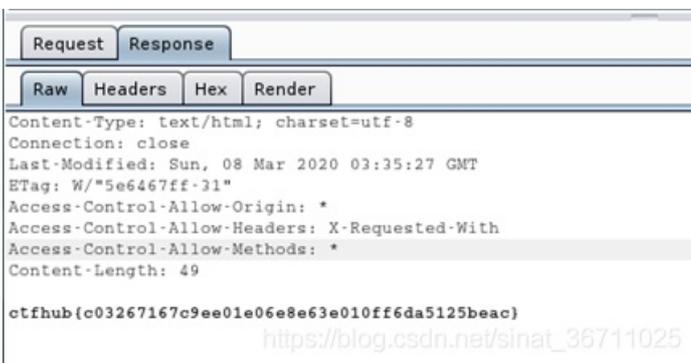
URL-encode these characters:

(5) Start Attack, 然后按 Length 排序, 并看到状态码出现200的, 即爆破成功

Request	Payload	Status	Error	Timeout	Length	Comment
22	YWRtaW46cXdlcnR5dWlvcA==	200	<input type="checkbox"/>	<input type="checkbox"/>	394	
0		401	<input type="checkbox"/>	<input type="checkbox"/>	404	
1	YWRtaW46MTIzNDU2	401	<input type="checkbox"/>	<input type="checkbox"/>	404	
2	YWRtaW46cGFzc3dvcmQ=	401	<input type="checkbox"/>	<input type="checkbox"/>	404	
3	YWRtaW46bGluZQ==	401	<input type="checkbox"/>	<input type="checkbox"/>	404	
4	YWRtaW46MTIzNDU2Nzg=	401	<input type="checkbox"/>	<input type="checkbox"/>	404	
5	YWRtaW46cXdlcnR5	401	<input type="checkbox"/>	<input type="checkbox"/>	404	
6	YWRtaW46MTIzNDU2Nzg5	401	<input type="checkbox"/>	<input type="checkbox"/>	404	
7	YWRtaW46MTIzNDU=	401	<input type="checkbox"/>	<input type="checkbox"/>	404	

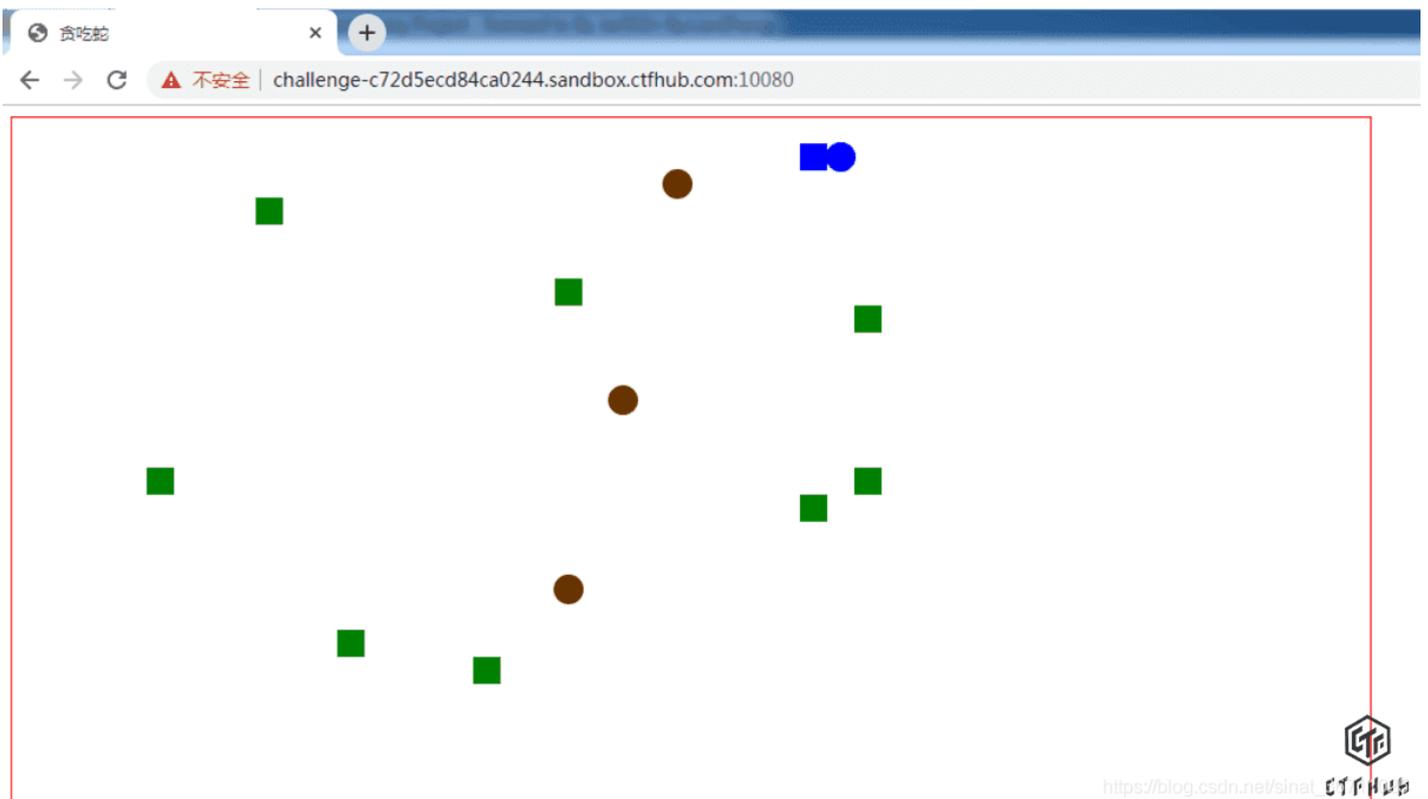


查看 Response, 得到flag



5、响应包源代码

题目页面打开后是一个网页版的“贪吃蛇”小游戏



根据题目要求是查看响应包源代码，可以使用burpsuite，也可以使用浏览器自带的开发调试工具，按f12进入调试界面，查找flag中的关键字段cfthub，就可以得到结果。

