

# [CTF刷题篇]CTFHub之Web前置技能HTTP协议Writeup(5/5)

原创

「已注销」于 2020-04-16 02:44:40 发布 696 收藏 1

分类专栏: [每日CTF # CTFHub](#) 文章标签: [安全](#) [web](#) [http](#) [python](#) [网络协议](#)

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/qq\\_43668710/article/details/105548987](https://blog.csdn.net/qq_43668710/article/details/105548987)

版权



[每日CTF](#) 同时被 2 个专栏收录

4 篇文章 0 订阅

订阅专栏



[CTFHub](#)

2 篇文章 0 订阅

订阅专栏

## 前言

CTFHub->Web->Web前置技能->HTTP协议->Writeup

### 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.

题目分析:

```
# Use CTF**B Method,Iwill give you flag  
也就是告诉你,用`CTF**B`的请求方式去访问就会给flag
```

姿势1:curl命令

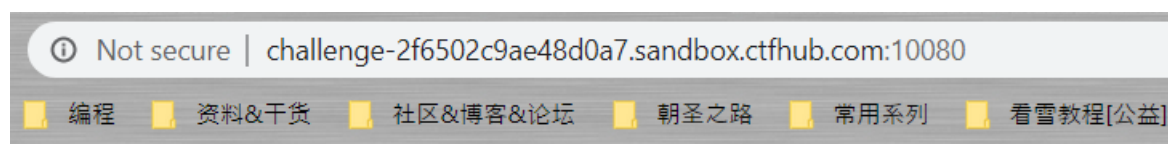
姿势2:burp

我这里选的curl命令,因为简单嘛

```
C:\Users\HP\Desktop
λ curl -v -X CTFHUB http://challenge-de28ea4be2ae5977.sandbox.ctfhub.com:10080/index.php
* Trying 47.98.148.7...
* TCP_NODELAY set
* Connected to challenge-de28ea4be2ae5977.sandbox.ctfhub.com (47.98.148.7) port 10080 (#0)
> CTFHUB /index.php HTTP/1.1
> Host: challenge-de28ea4be2ae5977.sandbox.ctfhub.com:10080
> User-Agent: curl/7.55.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: openresty/1.15.8.2
< Date: Wed, 15 Apr 2020 15:39:52 GMT
< Content-Type: text/html; charset=UTF-8
< Transfer-Encoding: chunked
< Connection: keep-alive
< X-Powered-By: PHP/5.6.40
< Access-Control-Allow-Origin: *
< Access-Control-Allow-Headers: X-Requested-With
< Access-Control-Allow-Methods: *
<
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8"/>
  <title>CTFHub HTTP Method</title>
</head>
<body>
  good job! ctfhub{691052d[REDACTED]f10176c1ce680ef3ee128a326d5}
</body>
</html>
* Connection #0 to host challenge-de28ea4be2ae5977.sandbox.ctfhub.com left intact
```

请求成功,flag回显在了窗口中

## 2.302跳转



# No Flag here!

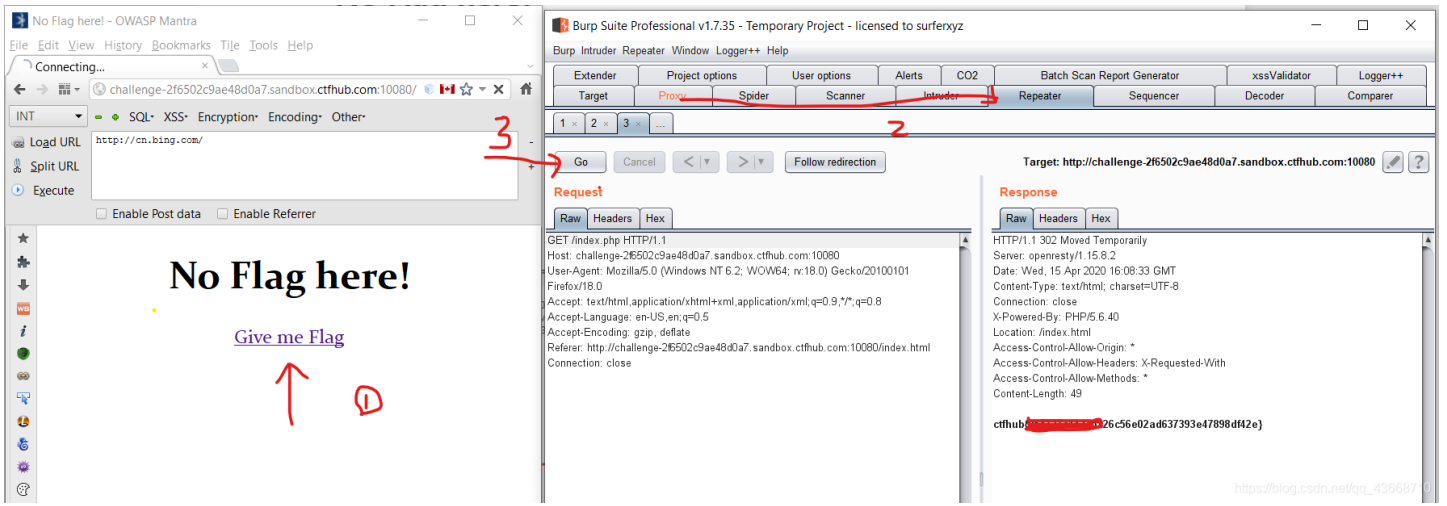
[Give me Flag](#)

[https://blog.csdn.net/qq\\_43668710](https://blog.csdn.net/qq_43668710)

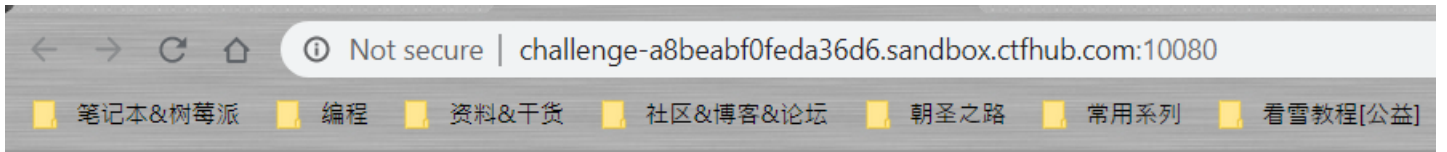
题目分析:

点击 "Give me flag"  
发现有302跳转  
于是可以想到用burp去抓包  
然后丢到repeater模块重发

最后在响应报文中发现flag:

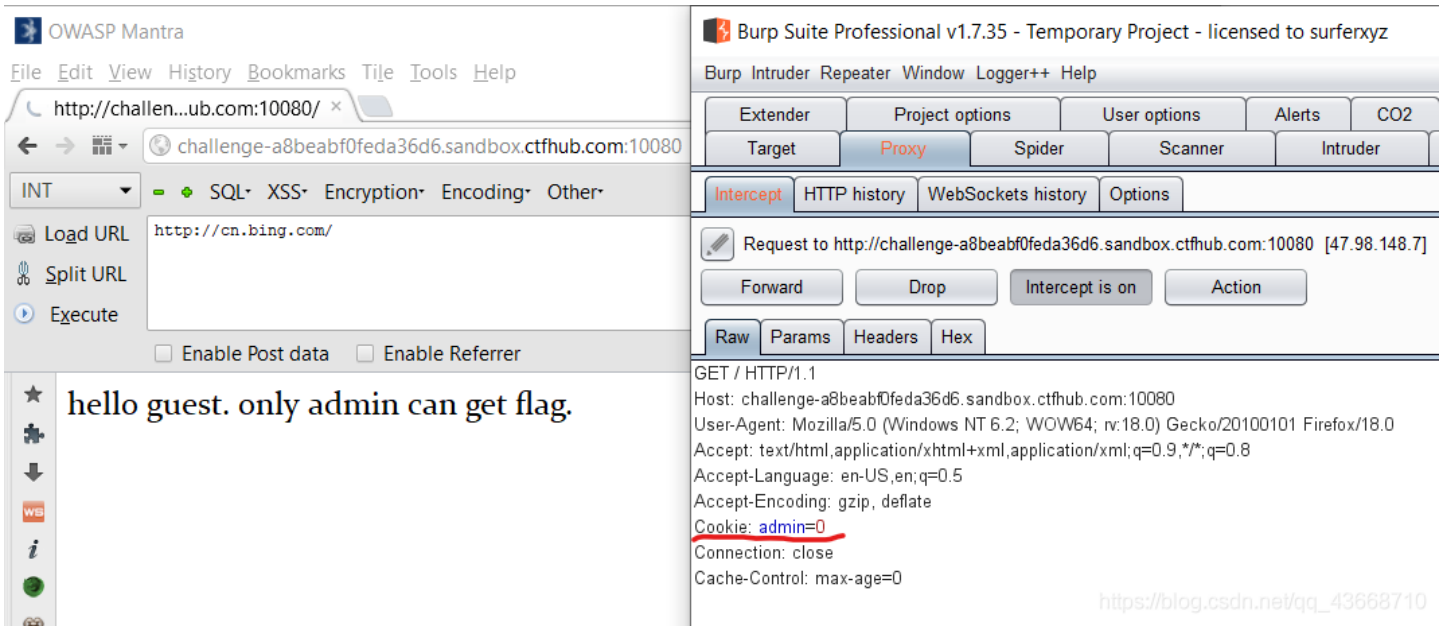


### 3.Cookie



hello guest. only admin can get flag.

发现自己权限不够,常规思路都是抓包,看看有没有可疑的参数



果然如我们想的一样

Cookie: admin=0

直接丢进repeater,把0改成1即可得到flag:

Request	Response
<pre>GET / HTTP/1.1 Host: challenge-a8beabf0feda36d6.sandbox.ctfhub.com:10080 User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64; rv:18.0) Gecko/20100101 Firefox/18.0 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 Cookie: admin=1 Connection: close Cache-Control: max-age=0</pre>	<pre>HTTP/1.1 200 OK Server: openresty/1.15.8.2 Date: Wed, 15 Apr 2020 16:31:00 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: 48  ctfhub{[REDACTED]d1f7716ac208a25d5183540a7ad8129fd636c}</pre>

[https://blog.csdn.net/qq\\_43668710](https://blog.csdn.net/qq_43668710)

## 4.基础认证

### HTTP基本认证(维基百科)

#### 简介

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，提供用户名和密码的一种方式。

在进行基本认证的过程里，请求的HTTP头字段会包含 `Authorization` 字段，形式如下：`Authorization: Basic <凭证>`，该凭证是用户和密码的组成的base64编码。

#### 电文认证过程

##### (1).客户端请求（没有认证信息）

```
GET /private/index.html HTTP/1.0
Host: localhost
```

##### (2).服务端应答

```
HTTP/1.0 401 Authorization Required
Server: HTTPd/1.0
Date: Sat, 27 Nov 2004 10:18:15 GMT
WWW-Authenticate: Basic realm="Secure Area"
Content-Type: text/html
Content-Length: 311

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/1999/REC-html401-19991224/loose.dtd">
<HTML>
  <HEAD>
    <TITLE>Error</TITLE>
    <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=ISO-8859-1">
  </HEAD>
  <BODY><H1>401 Unauthorized.</H1></BODY>
</HTML>
```

##### (3).客户端请求（有认证信息）

用户名“Aladdin”，密码，password “open sesame”

```
GET /private/index.html HTTP/1.0
Host: localhost
Authorization: Basic QWxhZGRpbjpvGVuIHNlc2FtZQ==
```

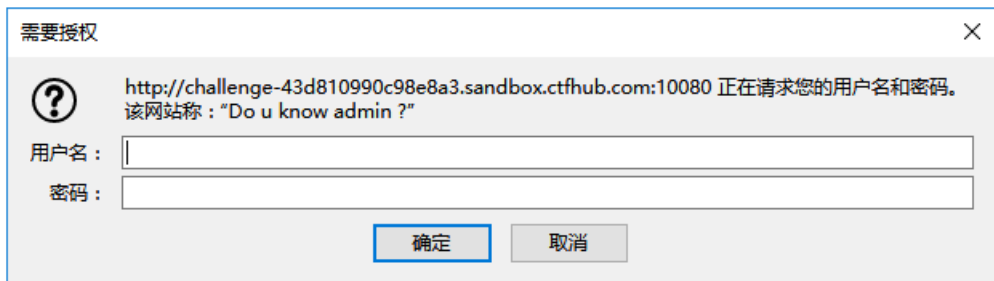
Authorization消息头的用户名和密码的值可以容易地编码和解码。

#### (4).服务端的应答

```
HTTP/1.0 200 OK
Server: HTTPd/1.0
Date: Sat, 27 Nov 2004 10:19:07 GMT
Content-Type: text/html
Content-Length: 10476
```

#### 题目分析

现在差不多明白基本认证的概念了,开始分析题目:



[https://blog.csdn.net/qq\\_43668710](https://blog.csdn.net/qq_43668710)

思路分析:

```
用户名:admin
附件:10_million_password_list_top_100.txt
说明我们需要爆破
用burp抓包,发现我们输入的密码被加密了
很明显是base64
把加密后的密文解密
发现是我们刚输入的密码和admin组合之后再行base64加密
```

所以现在就需要写一个脚本对附件中的密码批量进行base64加密

贴一个别人的脚本:

```

import base64

f=open('10_million_password_list_top_100.txt','r')
spring=open('test123456.txt','w')
lines=f.readlines()
for line in lines:
    Authorization="admin:"+line.strip('\n')
    Authorizationbs64=base64.b64encode(Authorization)
    print Authorizationbs64
    spring.write(Authorizationbs64)
    spring.write("\n")

f.close()
spring.close()

```

[https://blog.csdn.net/qq\\_43668710](https://blog.csdn.net/qq_43668710)

然后丢进burp的intruder模块爆破

The screenshot shows the Burp Suite Professional interface. On the left, a browser window displays the CTFHub challenge page with the text "CTFHub 基础认证" and "Here is your flag: [click](#)". The main window shows the Intruder tool configuration for a "Sniper" attack type. The "Payload Positions" tab is active, showing the configuration for inserting payloads into the "Authorization" header. A list of 16 payloads is shown, with the 42nd payload, "YWRtaW46YXNkZmdo", highlighted in orange. The "Results" tab shows the attack results, with the 42nd result being successful (Status: 200, Length: 354). A detailed view of the successful result is shown on the right, displaying the request and response. The response is an HTML page with a status of 200 OK and a content length of 49. The response body contains the flag: "ctfhub{172205d3e5591897d0b1cc10e895f61cb}".

爆破成功,访问正确的密码的数据包,即可得到flag!

## 5.响应包源代码

查看源码即可得到flag

```

<body>
  <canvas id="canvas" width="1000" height="700"></canvas>
  <div>
    <input id="switch" type="button" value="開始" onclick="clickSwitch()"></input>
  <br/>
    <input id="content" type="text" value="0"></input>
  </div>
</body>
<!-- ctfhub{172205d3e5591897d0b1cc10e895f61cb} -->
<script type="text/javascript">

```

结语

到此http协议模块就结束了.

返回上层

□ 未学习 ■ 学习中 ■ 已掌握



[https://blog.csdn.net/qq\\_43668710](https://blog.csdn.net/qq_43668710)