2021湖北省工匠杯预赛WriteUP

原创

TaibaiXX1 于 2021-08-19 11:32:14 发布 217 收藏
文章标签： web 数据可视化 vim html 信息安全

本文链接：https://blog.csdn.net/tangshuangsss/article/details/119814507

点击"仙网攻城狮”关注我们哦~

不当想研发的渗透人不是好运维 

让我们每天进步一点点

简介

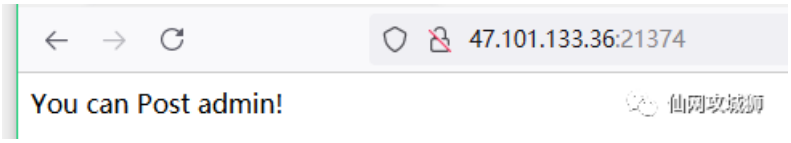
第一次参加CTF比赛目前预赛已经完结，预赛有6道题由于本人太菜只做出4道  ，感觉进决赛应该没有问题，本文分享解题思路

实战

WEB两题、其他的每种各一题

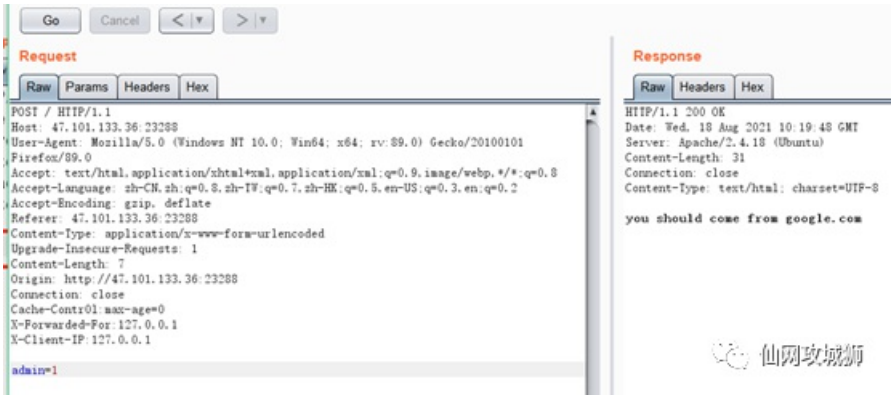
一、BabyWeb--绕过

1.打开题目说是使用post提交admin参数



2.构建POST参数后无果

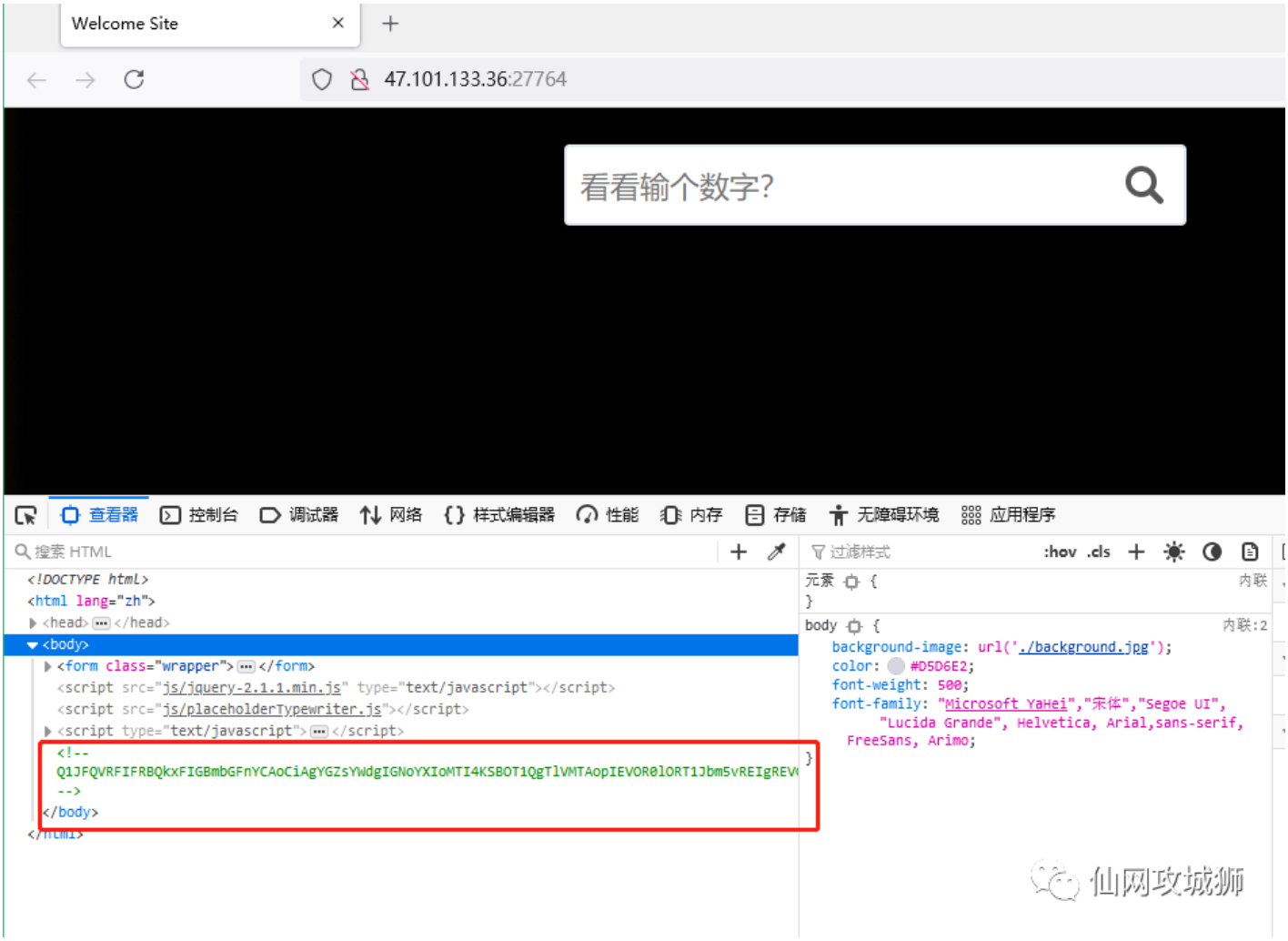
3.添加其他参数后提示需要把referer配置成www.google.com



4.修改后获得flag

二、Welcome_Site---SQL注入

1.打开题目看到有一串加密值，解密后发现表名和加密值长度

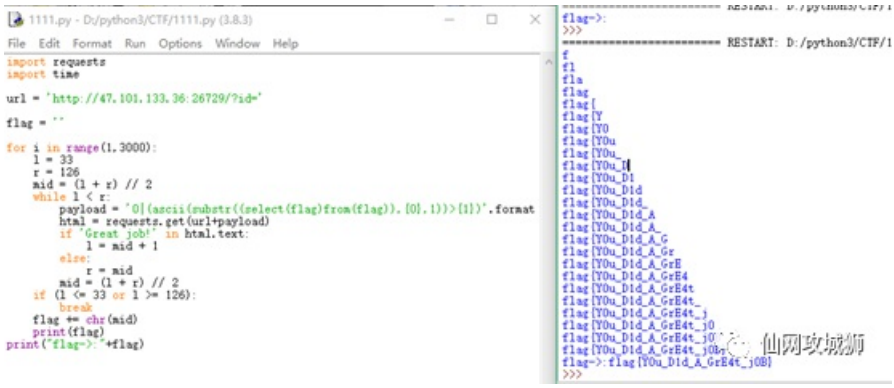


解密后

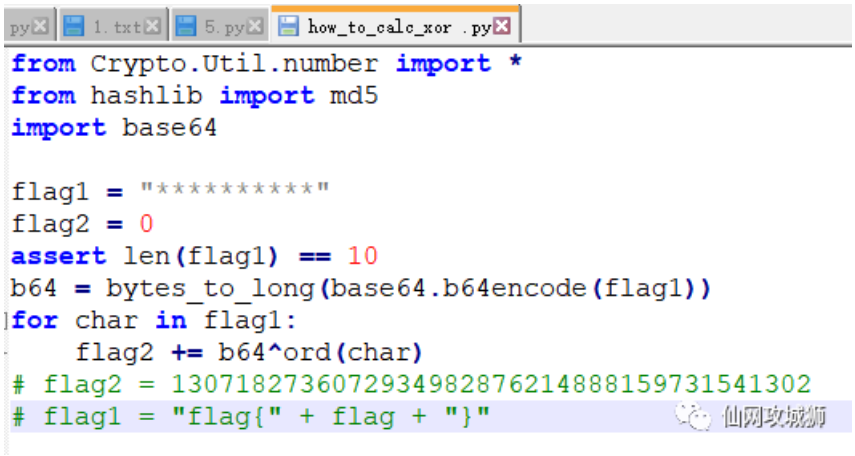
2.1-4可以显示，输入0发现返回nonono

3.尝试字符是否过滤发现|可以使用，显示1的界面字符，可以判断存在布尔型sql注入

4，编辑脚本跑出flag



三、how_to_calc_xor--密码学



1.这是道密码学的题，修改源代码分别打印仅base64和加上后面异或值的结果

```
└─$ cat bbb.py
from Crypto.Util.number import *
from hashlib import md5
import base64

def test(flag1):
        flag2 = 0
        assert len(flag1) == 10
        b64 = bytes_to_long(base64.b64encode(flag1))
        for char in flag1:
                flag2 += b64^ord(char)
        print b64
        print flag2


test("aaaaaaaaaa")
test("bbbbbbbbbb")

┌──(kali㉿kali)-[~/Desktop]
└─$ python2 bbb.py
118754449487600874703500827583755402557
118754449487600874703500827583754025880
118868761227381833308224926131511835965
118868761227381833308224926131511835990

┌──(kali㉿kali)-[~/Desktop]
└─$
```

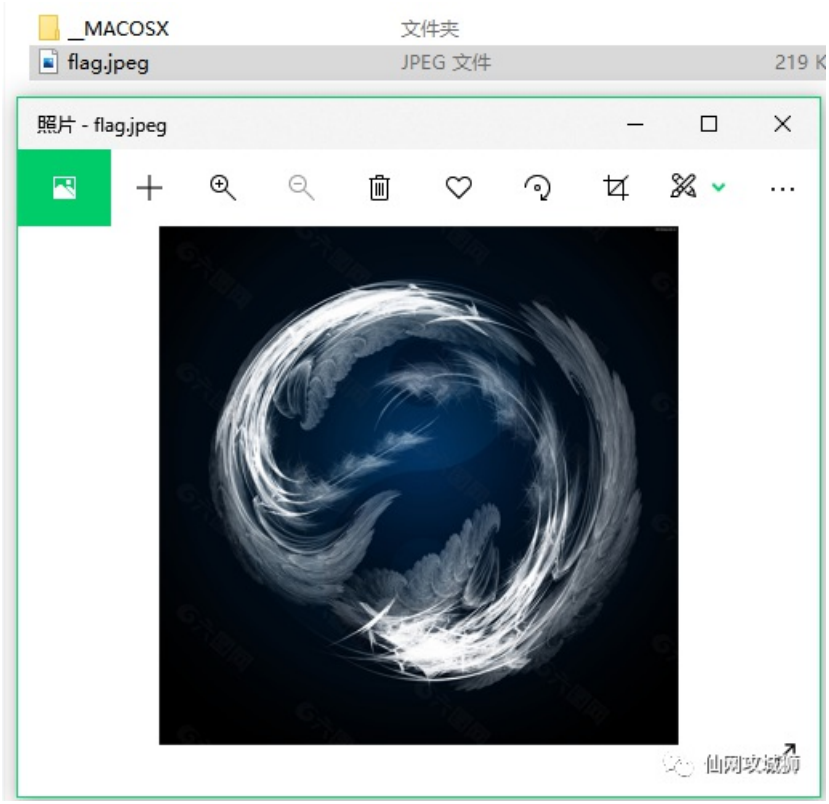

2.经过分析发现后比b64 和 flag2 区别在与去掉一位之后还有2位不一样，它是从33到127异或一个树可以先整体乘10然后再拼后面两位。

3.按这个规律修改13071827360729349828762148881597315 4xxx暴力破解flag1，得到flag

```
File  Actions  Edit  View  Help
└─$ vim ctf.py

┌──(kali㉿kali)-[~/Desktop]
└─$ python2 ctf.py
miscsoeasy
13071827360729349828762148881597315 4109

┌──(kali㉿kali)-[~/Desktop]
└─$ cat ctf.py
from Crypto.Util.number import *
import base64

for i in range(13071827360729349828762148881597315 4000, 13071827360729349828762148881597315 4999):
    try:
        res = base64.b64decode(long_to_bytes(i))
        print(res)
        print(i)
    except Exception as e:
        pass

┌──(kali㉿kali)-[~/Desktop]
└─$ python2 ctf.py
miscsoeasy
13071827360729349828762148881597315 4109

┌──(kali㉿kali)-[~/Desktop]
└─$
```


四、Guessing--图片隐写

1.开启题目后是一张图片



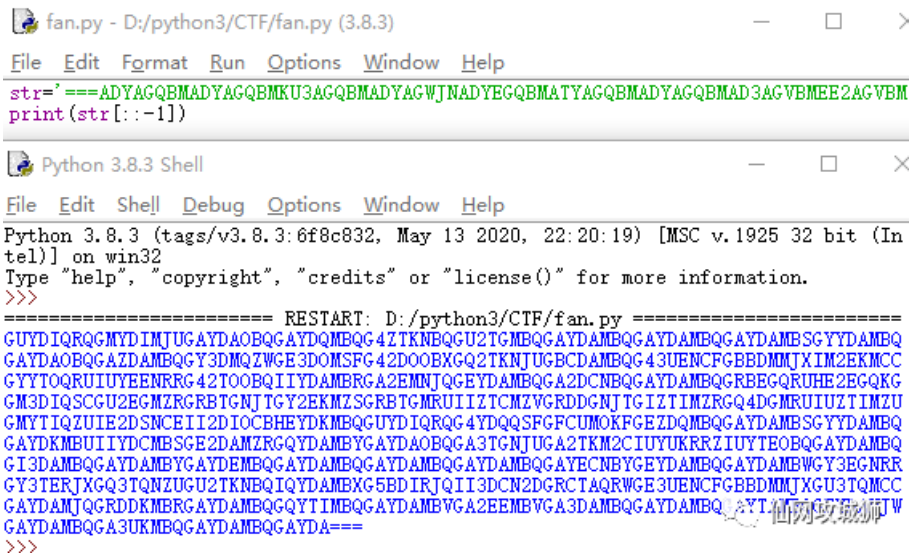
2.Hexdump 查看flag.jpeg，后面有一串加密字符，单独提取出来。

```
文件(F)  编辑(E)  查看(V)  搜索(S)  终端(T)  帮助(H)
00036f10  12 46 76 a0 db 61 60 34  12 10 ac 9e b8 db 34 0f  |.Fv..a`4......4.|
00036f20  09 44 76 ca 94 72 7a 9f  13 41 a6 e3 8a 75 5c 4a  |.Dv..rz..A...u\J|
00036f30  34 0c c8 f1 a6 d7 42 a6  cd 0a 8a 28 0a 02 80 a0  |4.....B....(....|
00036f40  28 0a 02 80 a0 28 13 8b  c2 8b a2 64 d4 db 44 a9  |(....(.....d..D.|
00036f50  b0 54 05 01 57 49 b1 5a  64 50 14 05 67 6d 68 13  |.T..WI.ZdP..gmh.|
00036f60  8a 9d a9 a4 e6 80 a0 28  0a 02 80 a0 28 0a 02 80  |.......(....(...|
00036f70  a0 28 0a 02 80 a0 28 0a  02 80 a0 28 0a 02 80 a0  |.(....(....(....|
00036f80  28 0a 0f ff d9 3d 3d 3d  41 44 59 41 47 51 42 4d  |(....===ADYAGQBM|
00036f90  41 44 59 41 47 51 42 4d  4b 55 33 41 47 51 42 4d  |ADYAGQBMKU3AGQBM|
00036fa0  41 44 59 41 47 57 4a 4e  41 44 59 45 47 51 42 4d  |ADYAGWJNADYEGQBM|
00036fb0  41 54 59 41 47 51 42 4d  41 44 59 41 47 51 42 4d  |ATYAGQBMADYAGQBM|
00036fc0  41 44 33 41 47 56 42 4d  45 45 32 41 47 56 42 4d  |AD3AGVBMEE2AGVBM|
00036fd0  41 44 59 41 47 51 42 4d  49 54 59 51 47 51 42 4d  |ADYAGQBMITYQGQBM|
00036fe0  41 44 59 41 47 52 42 4d  4b 44 44 52 47 51 4a 4d  |ADYAGRBMKDDRGQJM|
00036ff0  41 44 59 41 47 43 43 4d  51 54 33 55 47 58 4a 4d  |ADYAGCCMQT3UGXJM|
00037000  4d 44 42 42 47 46 43 4e  45 55 33 45 47 57 52 51  |MDBBGFCNEU3EGWRQ|
00037010  41 54 43 52 47 44 32 4e  43 44 33 49 49 51 4a 52  |ATCRGD2NCD3IIQJR|
00037020  49 44 42 35 47 58 42 4d  41 44 59 51 49 51 42 4e  |IDB5GXBMADYQIQBN|
00037030  4b 54 32 55 47 55 5a 4e  51 54 33 51 47 58 4a 52  |KT2UGUZNQT3QGXJR|
00037040  45 54 33 59 47 52 52 4e  47 45 33 59 47 57 42 4d  |ET3YGRRNGE3YGWBM|
```

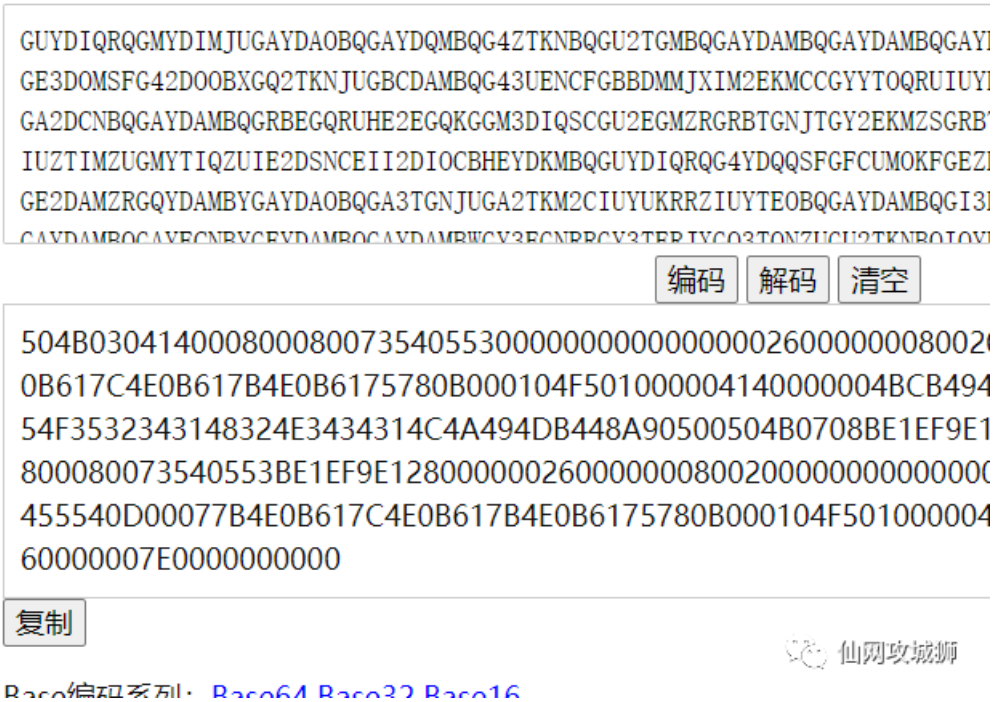
3.分析后发现有点像base32反码

```
root@kali:~# cat 111.txt
===ADYAGQBMADYAGQBMKU3AGQBMADYAGWJNADYEGQBMATYAGQBMADYAGQBMAD3AGVBMEE2AGVBMADYAG
QBMITYQGQBMADYAGRBMKDDRGQJMADYAGCCMQT3UGXJMMDBBGFCNEU3EGWRQATCRGD2NCD3IIQJRIDB5G
XBMADYQIQBNKT2UGUZNQT3QGXJRET3YGRRNGE3YGWBMADYAGQBMADYEGYBNCEYAGQBMADYAGQBMADYAG
QBMADYAGQBMEDYAGYBMADYAGQBMAD3IGQBMADYAGQBOETYUIZRRKUYUIC2MKT2AGUJNGT3AGQBOADYAG
YBMADYQGRZMAD2EGSBMCDYIIUBMKDYAGQBMADYYGSBMADYAGQBMQDZEGFKOMUCFGFSQQDY4GQRQIDYUG
QBMKDYEHBCOID2IIECNSD2EIUZQITYMGUZMITZUIURMGD4QGRZMITZIGTJNGDDRGVZMCTZIIURMGTBRG
SZMKE2YGTJNGTBRGRZMGE2UGCSQID3MGGKQGE2EHURQGEBRGQBMADYAGQBNCD2AGQBMADYEGQJNME2AG
RBMADYIIQBOOT24GRRNEEYUIURQOTYYGCCMKE2MIXJMMDBBGFCNEU34GQBMADCBGUJNKT2QGXBOOD24G
FSMOD3EGWZQMD3YGQBMADZAGQBOADYAGQBMADYYGSBMADYAGQBMADYAGQBMADYAGQBMADYAG
QBMQDYAGQBOADYAGUJMIDYMGQRQIDYUG
root@kali:~#
```

4.用python反转字符

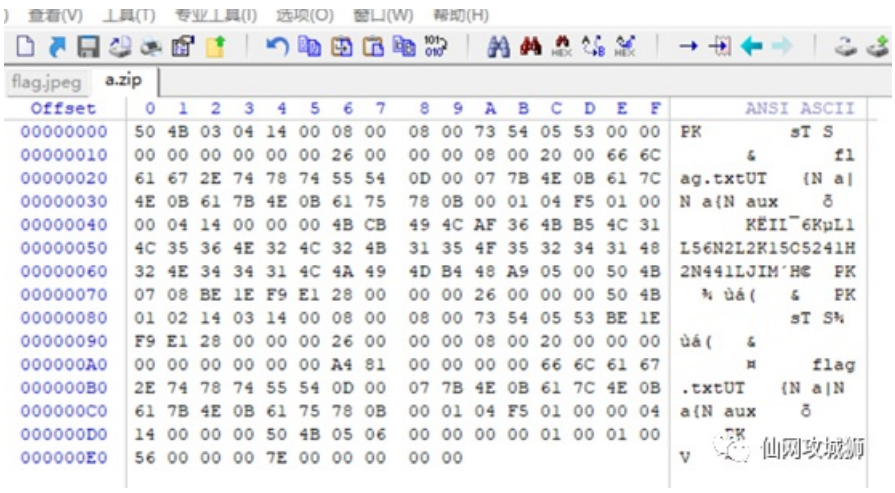


5.用base32解码

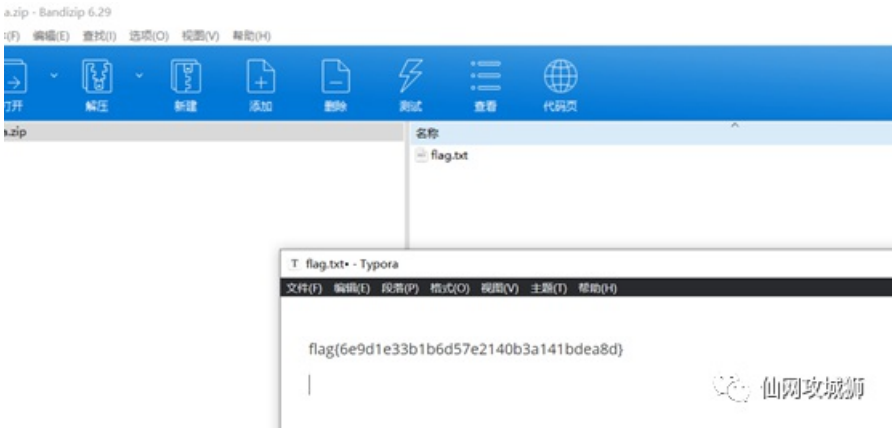


6.分析后发现是zip格式，16进制转换为字符串

7.使用16进制编辑工具生成ascii



8.保存为zip格式获取flag



总结：还是太菜了 

往期内容

CTFHub SSRF(服务器请求伪造) WriteUP

ATT&CK实战-红队评估之二

Web常见漏洞&逻辑漏洞学习文档分享

更多资讯长按二维码 关注我们

觉得不错点个“赞”呗