

## Crypto 50 我叫李二狗（一）

Base64 解密后，把 l 改成 1，x 和 o 改成 0，可以查到原文，主办方应该是替换了一些字母：

| 时间                  | 密文                               | HashType | 结果      |
|---------------------|----------------------------------|----------|---------|
| 2016-05-14 10:10:26 | 213fead602286cc06b70496b125ef22b | left16   | We1c0me |

替换前结果：213fead602286cc06b7d496b125ef22b

替换后结果：2l3fead6o2286cco6b7x496bl25ef22b

(anyway.....)

Flag: whctf{We1c0me}

## Crypto100 李二狗的梦中情人

图片尾部有另外一个链接，下载到一张相似的图片，然后根据提示找不同 sub 一下得到二维码，用 PS 反色后扫描可以得到 flag（当时扫的时候没有注意 flag 中间有空格.....）：



Flag: whctf{hel10 ber7@hust\_is}

## Crypto300 我叫李二狗（二）

题目给了一个双层 RSA 的公钥文件，首先使用 openssl 提取 n 和 e：  
openssl rsa -RSAPublicKey\_in -in [RSAPUBKEY 文件名] -text -modulus  
得到 n1 和 n2 后直接丢到 factordb 分解，得到对应的 pq，写脚本解密：

```
def main():
    c = 696071995178684833591654466965757829867984913331716345614800719762341435383453672347994151191243721015386803107689873818463070942531699822964664774070404247696

    n1 = 10004104526045228399
    p1 = 973386913
    q1 = 10277623823
    e1 = 0x10001

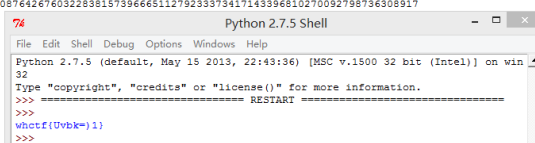
    d1 = gcd(e1, (p1-1)*(q1-1))

    n2 = 12301866845301177551304949583849627207728535695953347921973224521517264005072636575187452021997864693899564749427740638459251925573263034537315482685079170261
    p2 = 3347807169895698786044169848212690817704794983713768566912431388982883793878002287614711652531743087737814467999489
    q2 = 36746043666799590428244633799627952632279158164943087642676032283815739666511279233373417143396810270092798736308917
    e2 = 0x10001

    d2 = gcd(e2, (p2-1)*(q2-1))
    m1 = pow(c, d1, n1)
    m2 = pow(m1, d1, n1)

    print 'whctf{' + hex(m2)[2:-1].decode('hex') + '}'

if __name__ == '__main__':
    main()
```



Flag: whctf{Uvbk=)1}

## Crypto200 李二狗的 LOL 战歌

插上耳机的时候发现有个声道里面是摩斯电码。记录下来后是

..-./--/...--/-./...-/...--/...--/-...-/...-/...--/...--

解密之后是 fg3trv22bun7q

然后 rot13 后得到 st3gei22oha7d。

Flag: whctf{st3gei22oha7d}

## Reverse100 小菜一碟

对用户的输入做了个加密运算

运算方式很简单

```
11 seeda = seed;
12 result = (int *)malloc(0x64u);
13 lenString = strlen(string);
14 lenKey = strlen(key);
15 for ( count = 0; count < lenString; ++count )
16 {
17     v = string[count];
18     k = key[seeda];
19     if ( checkIn(string[count], key) )
20         result[count] = (16 * v + k) % 2500;
21     else
22         result[count] = (v ^ (k << seeda)) % 2500;
23     seeda = (seeda + 5) % lenKey;
24 }
25 return result;
26 }
```

其中 key 是一串字符串 key='Just try your best and enjoy yourself!'

Python 重新写一下算法，一位一位的爆破即可算出 flag:

```
E:\>python rel.py
0 w w
1 h wh
2 c whc
3 t whct
4 f whctf
5 { whctf{
6 y whctf{y
7 o whctf{yo
8 u whctf{you
9 _ whctf{you_
10 a whctf{you_a
11 r whctf{you_ar
12 e whctf{you_are
13 _ whctf{you_are_
14 t whctf{you_are_t
15 h whctf{you_are_th
16 e whctf{you_are_the
17 _ whctf{you_are_the_
18 b whctf{you_are_the_b
19 e whctf{you_are_the_be
20 s whctf{you_are_the_bes
21 t whctf{you_are_the_best
22 } whctf{you_are_the_best}
whctf{you_are_the_best}
```

代码:

```





#coding=utf-8
ss=[0x009D,0x960,0x8A7,0x7B3,0x6C4,0x5F7,0x805,0x756,0x7C4,0x313,0
x630,0x740,0x6B5,0x88B,0x7B3,0x022,0x6C4,0x4EB,0x685,0x6BE,0x79A,0
x7AF,0x559]
key='Just try your best and enjoy yourself!'
rst=''
for j in range(23):
    for i in range(33,126,1):
        t=rst+chr(i)
        b=t.ljust(23,'A')
        result=[]
        lenkey=len(key)
        seed=1
        for index in range(len(b)):
            v=b[index]
            k=key[seed]
            if(b[index] in key):
                tmp=(16*ord(v)+ord(k))%2500
                result.append(tmp)
            else:
                tmp=(ord(v)^(ord(k)<<seed))%2500
                result.append(tmp)
            seed=(seed+5)%lenkey
        if result[j]==ss[j]:
            rst=t
            print j,chr(i),t
            break
print rst

```

## Reverse200 CrackMe

QT 程序，试了半天没有办法运起来，只好 IDA 静态看算法，还好不复杂。避免有坑，用 QT 还原下逻辑，运行得到注册码。

有个坑点：

|   |                             |   |  |
|---|-----------------------------|---|--|
|  | .rdata:00000000... 00000007 | C | 确定   |
|  | .rdata:00000000... 00000052 | C | 请提交whctf(\$serial),serial为破解所得值。欢迎交流szxwlp@foxmail.com |
|  | .rdata:00000000... 00000006 | C | name:  |
|  | .rdata:00000000... 00000008 | C | Serial:  |

一直以为是序列号加上 whctf{}，后来才发现还要加上\$，==

PHP 程序员表示不服……

```

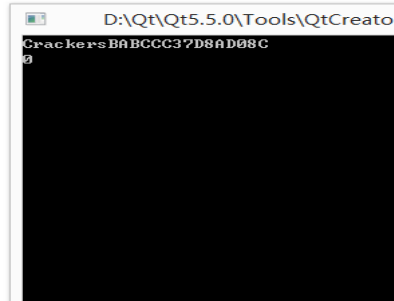
unsigned int v2 = 0xAAAAAAAA;
int v8;
int l = strlen(tmp);
for (i=0; i<=l-1; i++)
{
    BYTE v7 = *((BYTE *)tmp + i);
    if (i % 2)
    {
        v8 = ~((v2 << 11) + (v7 ^ (v2 >> 5)));
    }
    else
    {
        v8 = (v2 << 7) ^ (v7 * (v2 >> 3));
    }
    v2 ^= v8;
}

QString str2;
str2 = QString::number(v2, 16);
str2 = str2.toUpper();
v18.append(str2);

int len = v18.length();
int j;
QByteArray qb2 = v18.toLatin1();
char *tmp2 = qb2.data();

int v3=0;
for (j=0; j<=len-1; j++)
{
    v3 = qb2[j] + 0x83 * v3;
}

```



Flag: whctf{\$CrackersBABCCC37D8AD08C}

## Reverse200 我讨厌数学

这个算法动态调一下就能明白，把输入的 flag 放到一个 6\*6 的矩阵 A 里面。然后给定常量矩阵 B，求：

$$A^T * A = B$$

其中 B = [[73767, 62102, 48910, 55372, 37336, 663], [62102, 55344, 41766, 45072, 30247, 560], [48910, 41766, 36843, 34717, 28867, 445], [55372, 45072, 34717, 45069, 28239, 503], [37336, 30247, 28867, 28239, 39900, 348], [663, 560, 445, 503, 348, 6]]

题目中还告诉你，flag 长度为 27 位，剩下 9 位补 1，于是根据 flag 格式可以得到：

$$A = \begin{bmatrix} w & h & c & t & f & \{ \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12} \\ x_{13} & x_{14} & x_{15} & x_{16} & x_{17} & x_{18} \\ x_{19} & x_{20} & \} & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

首先利用第五行和第一行、第六行的乘积，可以解出第五行最左边为 th。

然后第四行有 6 个未知数，但是只有 4 个方程，因此需要脑洞出另外两个变量，当时是猜出了倒数第三个是 '\_'、倒数第一个 '4'。然后利用 matlab 解方程，得到第四行。

然后第三行只需要猜出下划线的位置，第二行有 6 个方程，可以直接解出。

## Reverse 200 坐标定位

下载 apk 后发现这题和 Ali 之前移动安全挑战赛的题目基本一样。。。

对比了一下 Jeb 反编译后的代码，只在原题的基础上做了些小改变，思路完全一样

apk 会在 log 里输出转换表：

|               |       |       |                |     |  |
|---------------|-------|-------|----------------|-----|--|
| I 05-15 17... | 10375 | 10375 | com.example... | l1l | table:一乙二十丁厂七卜人八儿九儿了乃刀又三手干戈士工土才寸下大丈与万上小口 |
| I 05-15 17... | 10375 | 10375 | com.example... | l1l | 井开夫天无元专云扎艺术五支序不太大区历尤友匹丰巨牙屯比互切瓦止少日申冈贝内水见  |
| I 05-15 17... | 10375 | 10375 | com.example... | l1l | 文六方火为斗忆订计户认心尺引丑巴孔队办以允手勋双书约玉刊示来未击打巧正扑机功扔  |
| I 05-15 17... | 10375 | 10375 | com.example... | l1l | 只央兄叨叫另叨叹四生失禾丘付仗伐仙们仪自仔他斥瓜乎丛令用甩印乐          |
| I 05-15 17... | 10375 | 10375 | com.example... | l1l | pw:亡丸凡亡丸凡义义凡么么门凡广义凡之                     |
| I 05-15 17... | 10375 | 10375 | com.example... | l1l | enPassword:么广亡门义之尸号己丸凡勺夕个久及己口            |

最有要达到的结果是让 `enPassword == pw`

输入 1234567890+/\*,.;# 查看一下 log 看 enPassword 转换了什么，如上图

于是得到一个字符->汉字的对应关系

pw:亡丸凡亡丸凡义义凡么么门凡广义凡之转为字符后为

30-30-55-114-25-6

Flag: whctf{30-30-55-114-25-6}

## Reverse300 来华科

算法依旧很容易懂，但是程序没有输入，给了一个密文、一个明文和一个密钥，明文根据密钥加密后等于密文就是输出 bingo。

主办方后来提示 7 位数字，猜测是需要根据密文，在 7 位数字的范围内反推明文，然后把明文用来解密 zip。（加密 zip 藏在题目图片中……）

先写解密脚本：

```
def decrypt(key):
    a = 'bc160e612<f:cibg61e4035=gh71e03j'

    ss = ''
    k = key
    l = len(k)
    for i in range(0, 32):
        if ord(a[i]) >= ord('0') and ord(a[i]) <= ord('='):
            ss += chr(ord(a[i]) ^ (ord(k[i%l]) - 0x30))
        else:
            ss += chr(ord(a[i]) - (ord(k[i%l]) - 0x30))
    return ss
```

爆破得到密码为 1144305，然后解出 hello.pyc，用 uncompile 还原得到 py。接下来是脑洞环节，给了一个乱序的字符串和六个变换函数。自己读懂了功能后，组合出了一个，但是并不对（摔……）：

```
8f3@{ftchw52f45b4bbc?03e32d4!25adc97eysaeos}7
f3@{ftchw52f45b4bbc?03e32d4!25adc97eysaeos}78
whctf{@3fbb4b54f254d23e30?ce79cda52!87}soeasy
```

打算把这 32 位扔去解密试试的时候，瞅了一眼密钥，发现都在 0-5 之间……脑洞开了下，按照密码的方式组合了一次：

```
if __name__ == '__main__':
    d = 'f8@3f{ctwh254fb5b4cb0?e3234d2!a5cd79yeasoe}s7'
    #d = '1234567890abcdefghijklmnopqrstuvwxyzABCDEFGHI'
    data = []
    for i in range(len(d)):
        data.append(d[i])

    data = func1(data)
    print func6(data)
    data = func1(data)
    print func6(data)

    data = func4(data)
    print func6(data)

    data = func4(data)
    print func6(data)

    data = func3(data)
    print func6(data)

    data = func0(data)
    print func6(data)

    data = func5(data)
    print func6(data)
```

得到:

```
8@3f{ctwh254fb5b4cb0?e3234d2!a5cd79yeasoe}s7f
@3f{ctwh254fb5b4cb0?e3234d2!a5cd79yeasoe}s7f8
f{@3whct4f25b4b50?cb23e32!4dcda5ye79oeas7f}s8
@3f{ctwh254fb5b4cb0?e3234d2!a5cd79yeasoe}s7f8
3@{ftchw52f45b4bbc?03e32d4!25adc97eysaeos}f78
8@{ftchw52f45b4bbc?03e32d4!25adc97eysaeos}f7
whctf{t@38bb4b54f254d23e30?ce79cda52!7f}soeas7
```

Flag: whctf{@38bb4b54f254d23e30?ce79cda52!7f}

## Web100 beat it

一个只有三个包的 pcap 文件，按顺序把 data 部分粘到一起：

00350035002845b0303030303030303030303030373736383633373436363742333033  
3030303742333033303330323036383033303230363836353732363532303639373432303639  
373332303734363836353230363636433631363732303330333033303744

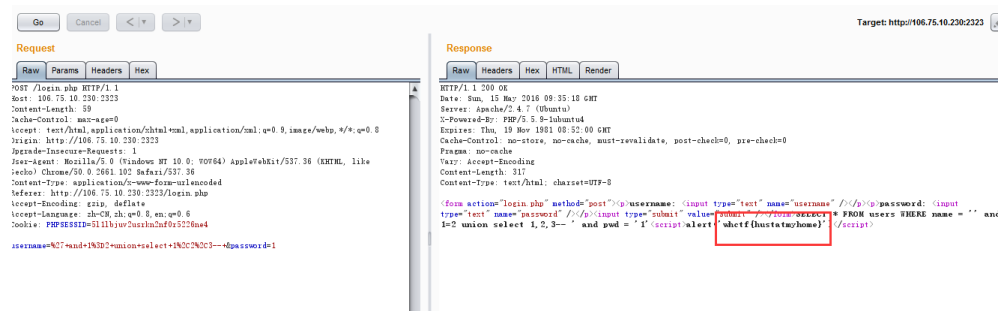
去掉开头的 0035，16 进制转 ascii，得到另一串 16 进制数，去掉开头的 0 和中间的 0，再次解码得到：

```
whctf{00{000 h0 here it is the flag 000}
```

看起来中间有重复，去重得到：

```
whctf{000 here it is the flag 000}
```

## Web100 忘了账户和密码



Payload: ' and 1=2 union select 1,2,3 --

Flag: whctf{hustatmyhome}

## Web100 find

找到这里：

```
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8" />
  <title>新闻网站</title>
  <link href="css/base.css" rel="stylesheet" type="text/css" />
  <link href="css/index.css" rel="stylesheet" type="text/css" />
  <link href="css/[a-zxhyz]{2}ctf{0-9}{7}.css" rel="stylesheet" type="text/css" />
</head>
<body>
```

然后猜测是 whctf+7 位数字，但是爆破可能太多了……后来主办方放了提示，给了一个 QQ 号（WTF？！），然后一大脑洞队友输入该 QQ 的生日 7 位数得到 css 中的 flag……

## Web200 窃取

看上去像个 sqli，丢给 sqlmap，从 flag 表中找到：  
flag is nothere,but I can tell you the flag is xor user's password.

读出两个 user 的密码：

8FC63BC4337CD4B5F70577118BB69FE8

6a3fba70c97c880679a740669ddd5ca3

异或：

e5f981b4fa005cb38ea23777166bc34b

解 md5 得到

hust

提交 flag: whctf{hust}

## Web200 密码忘了怎么办

又是个 sqli，没有回显的地方，时延盲注慢慢跑～

```
+-----+
| flag  |
+-----+
| *^sd  |
| --    |
| fwqe  |
| hello$$$|
| itisme&---&&&|
| trw\\  |
| user21123|
| whctf{ |
| wjiedwd|
| wnjkasdbnvkbd|
| }      |
+-----+
```

跑出整个 flag 列的数据，发现 whctf{,在大括号内对剩余字段排列组合提交无果。  
慢慢去跑其他列数据的时候发现已经有队伍提交成功了，于是排列组合再交一遍。。。

发现正确 Flag:

whctf{hello\$\$\$itisme&---&&&}

## Web200 信息

这个真不是 web 题目……

把带密码的压缩包下载下来，然后根据提示，输入密钥 [www.hust.edu.cn](http://www.hust.edu.cn) 就可以解压，  
然后看到 docx 里面的 flag……不明白和源码有啥关系。

## Web300 看图说话

.....

首先根据提示，curl 一下 ctf.php:

```
root@localhost:~/Desktop# curl 106.75.10.230:5566/ctf.php
<!DOCTYPE html>
<html>
<head>
<title>CTF</title>
<link rel="stylesheet" type="text/css" href="templates/standard/css/style_main.p
hp"/>
<link rel="stylesheet" type="text/css" href="templates/standard/css/style_form.c
ss"/>
</head>
<body>

```

然后访问下载图片看到 flag:

```
? ( àŽžô=ÇâE g""<ÑE p3@É Q@ Ö" ÔQ@ ÜÐ9Í P i13
flag.txt °È "whctf{today@@isnot09#$tomorrow}Ä={ @
```