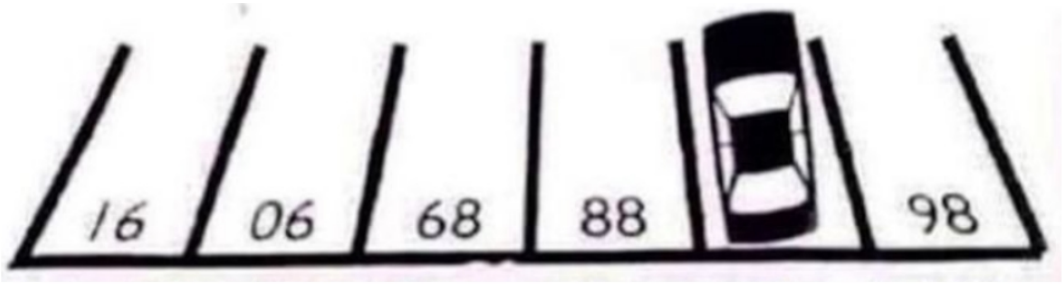
# XJNU-CTF 2018 WP

## BASE

### 0x01 Base10

题目如下：


打开网页看到一张图片：



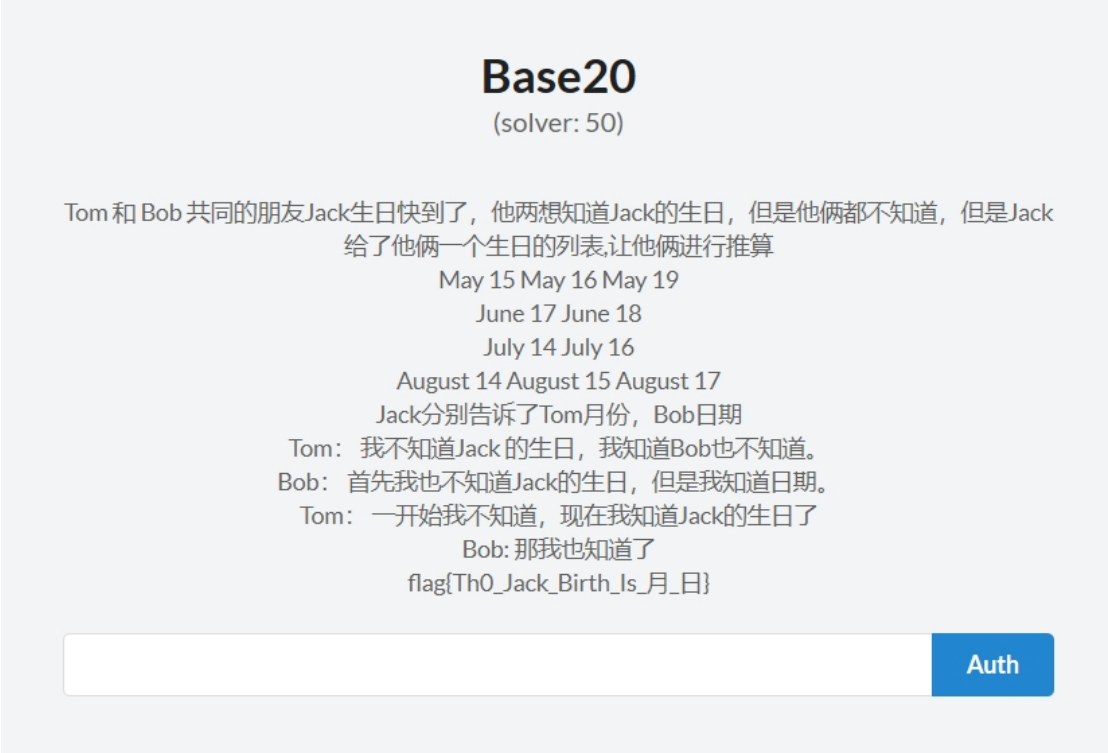

挡住的肯定是 87 啊，从右往左看就明白了，右键打开网页源代码：

```
</head>
<body>
<img src="base1.png">
<!-- flag{Math_is_Here_$num$!} 鑲板砬绔嬉劫琚▣皽杞￢溅鎸' 緎浜,num=md5(num)[:5]-->
</body>
</html>
```

将 87 进行 MD5 加密取前五位即可。

**0x02 Base20**

题目如下：




我智商不够，答案是一个一个试出来的。

**0x03 Base30**

题目如下：


直接上脚本：

```
a=1000000000-1

sum=0

while(a):

    if a%3==0 or a%5==0:

        sum+=a

    a-=1

print sum
```

**0x04 Base40**

题目如下：


直接 16 进制转字符串获得 flag

**0x05 Base 50**

题目如下：


代码如下：

```
import string

import itertools

import hashlib
```

```
enc='5bc76f3f319865431dcab801bbce47a1'

code = ''

strlist = itertools.product(string.letters + string.digits, repeat=4)


for i in strlist:
    code = i[0] + i[1] + i[2] + i[3]
    key='xjnu66'+code+'88ctf'
    encinfo = hashlib.md5(key).hexdigest()
    if encinfo == enc:
        print key
        Break
```

## MISC

### 0x01 Misc10

题目如下：


关注微博后，私信发送 flag

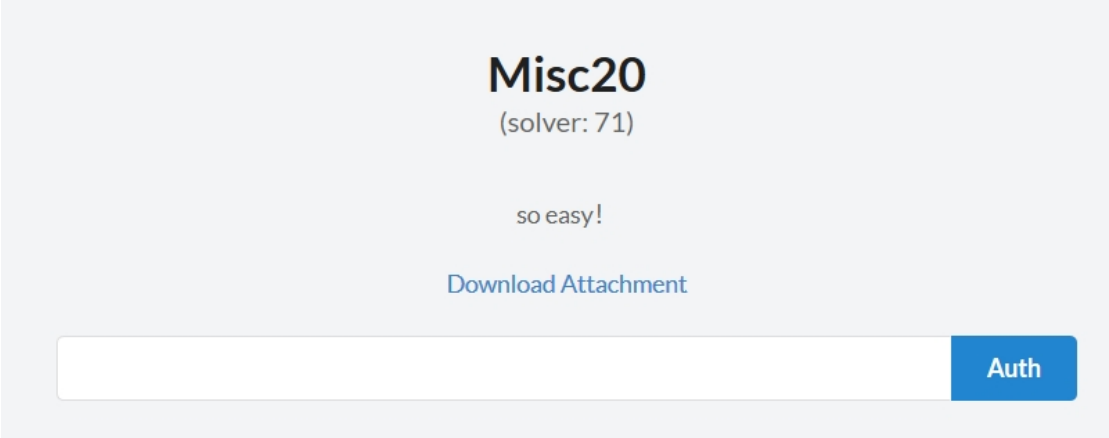
**0x02 Misc15**

题目如下：


将图片直接右键记事本打开，看到了 flag

**0x03 Misc20**

题目如下：




将下载的 misc10 直接 foremost，分离的图片即为 flag

**0x04 Misc40**

题目如下：


将下载好的 docx 文件，后缀名改为 zip，解压缩后即可看到以 flag 作为名字的文件夹。

## Crypto

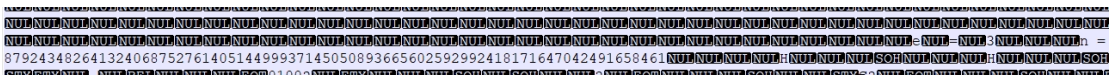
### 0x01 Crypto30

题目如下：


是一道简单的 rsa：

```
    </div>
</body>
<!-- hint m=587681053161488419997773704121869360186254866685321341947615498845105993905 92 -->
</html>
```

n 在图片里，e 为图片名，e=3：

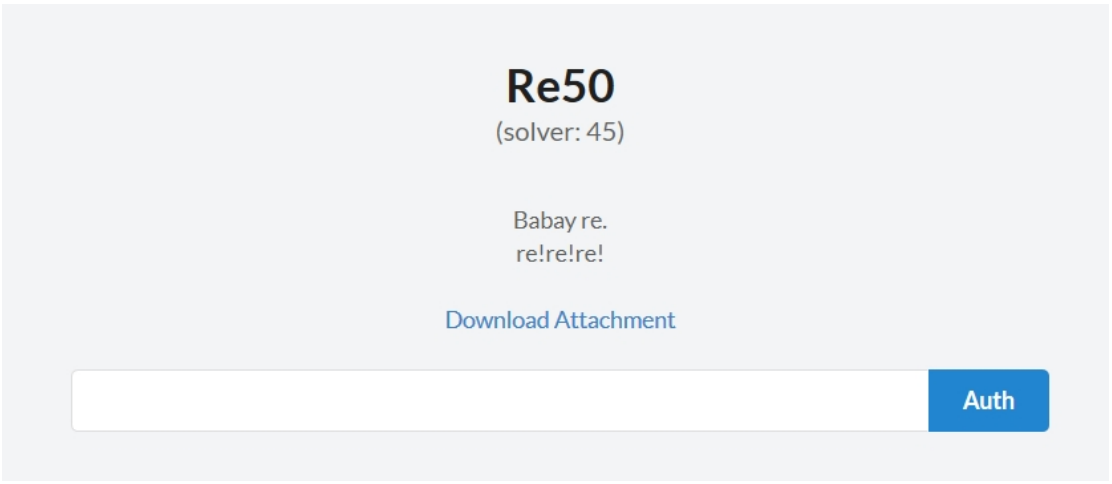


有了 n,e,c,直接分解 n，之后求出 d，最后解出明文

## RE

### 0x01 Re50

题目如下：




直接扔进 IDA，看下字符串：

```
461             mov     ebp, esp
463             and     esp, 0FFFFFFF0h
466             sub     esp, 30h
469             call    ___main
46E             mov     dword ptr [esp], offset aHiThisIsABabyr ; "Hi~ this is a babyre"
475             call    _printf
47A             mov     byte ptr [esp+2Fh], 66h
47F             mov     byte ptr [esp+2Eh], 6Ch
484             mov     byte ptr [esp+2Dh], 61h
489             mov     byte ptr [esp+2Ch], 67h
48E             mov     byte ptr [esp+2Bh], 7Bh
493             mov     byte ptr [esp+2Ah], 52h
498             mov     byte ptr [esp+29h], 65h
49D             mov     byte ptr [esp+28h], 5Fh
4A2             mov     byte ptr [esp+27h], 31h
4A7             mov     byte ptr [esp+26h], 73h
4AC             mov     byte ptr [esp+25h], 5Fh
4B1             mov     byte ptr [esp+24h], 53h
4B6             mov     byte ptr [esp+23h], 30h
4BB             mov     byte ptr [esp+22h], 5Fh
4C0             mov     byte ptr [esp+21h], 43h
4C5             mov     byte ptr [esp+20h], 30h
4CA             mov     byte ptr [esp+1Fh], 4Fh
4CF             mov     byte ptr [esp+1Eh], 4Ch
4D4             mov     byte ptr [esp+1Dh], 7Dh
4D9             mov     eax, 0
4DE             leave
```

这个就是 flag