

队伍 ID: 弱鸡肥宅 BBChan

队长: 孙博远 学号: 17130120188 QQ: 2951209135

队员: 孙燮阳 学号: 17130120193 QQ: 411588248

解题人:

1.welcome : 孙博远

2.nazo: 孙燮阳

3.我的世界: 孙博远

4.see or not see 孙燮阳

5.Moe: 孙博远

6.pacp: 孙博远

7.贪吃蛇: 孙燮阳

8.easy rsa : 孙博远

## Welcome 类型: MISC

签到题, 直接复制 flag 提交

Welcome

112

80 支队伍已解出

MiniLCTF{Welcome\_to\_MakerCTF233}

Flag

提交

## Nazo 类型: MISC

第一关: 签到关, 答案题面已给出 welcome

第二关: 点击 key 超链接, 即跳出答案 gotcha

第三关: 题面说 从右往左念, 下面有“where is key”, 则尝试输入“where”, 即答案

第四关: 是一句谚语, 搜索引擎搜一下即得到答案 survival

第五关: 题面说了 morse, 即下面的点线是摩尔斯电码, 在线转换工具得到答案: sos

第六关: 看出的 base64 加密, python 有 base64 库, 写一个解密脚本即可, 得到答案: 1029174037

第七关: 在 qq 上添加上一题答案 1029174037 为好友, 看到添加好友的问题三上有写 key: macintosh

第八关: 查询题面 IDNs 可知是: 国际化域名的简称, 将错的是: 世界复制粘贴到浏览器地

址栏上转到，即发现答案：saionjisekai

第九关： 题面为角度，将原图扔到 ps 里，横向拉长可看到单词：pineapple

第十关： 谷歌或百度搜索该图片，可知图片为早期鼠标，输入 mouse 即答案

第十一关：辨认出英文 nweroslesstofu

第十二关：数字游戏，A 表示出现过位置也对，B 表示出现过但位置不对，得到答案 9506

第十三关：查看网页源代码，打开对应的 png 文件，得到答案：thealpha

第十四关：在源代码里下载图片，利用 binwalk 发现是一个压缩文件，改后缀为 rar，得到压缩包里是一个 torrent 种子文件，下载该种子文件，得到答案 greendamn

第十五关：下载音频文件，用 Audacity 打开，调成频谱图发现是一张图片，在把图片翻转，得到答案：koenokiseki

第十六关：在源代码中打开 svg 图片，利用 F12 删掉改在 key 上面的块得到答案 secretvg

第十七关：打开网页源代码，发现中间的空白由两种不同的空格组成，复制到 Word，把圆角空格换成●，半角空格不动，得到如下形状：

A 5x10 grid of dots forming a stylized letter 'A'. The dots are arranged as follows (row by row):  
 Row 1: (1,1)-(1,4), (1,6), (1,7), (1,8)-(1,10), (1,12), (1,13), (1,14), (1,15), (1,16), (1,17), (1,18), (1,19), (1,20)  
 Row 2: (2,1), (2,2), (2,3), (2,4), (2,6), (2,7), (2,8), (2,9), (2,11), (2,12), (2,13), (2,14), (2,16), (2,17), (2,18), (2,19)  
 Row 3: (3,1)-(3,4), (3,6), (3,7), (3,8), (3,9), (3,11), (3,12), (3,13), (3,14), (3,16), (3,17), (3,18), (3,19), (3,20)  
 Row 4: (4,1), (4,2), (4,3), (4,4), (4,6), (4,7), (4,8), (4,9), (4,11), (4,12), (4,13), (4,14), (4,16), (4,17), (4,18), (4,19)  
 Row 5: (5,1)-(5,4), (5,6), (5,7), (5,8), (5,9), (5,11), (5,12), (5,13), (5,14), (5,16), (5,17), (5,18), (5,19), (5,20)

因此 entropy 即为答案，也是本题 flag

See or not see      类型: MISC

下载题目对应的 PDF 文件, 用 WPS 打开, 发现自动分成了三个图层, 第二个图层上即有 flag

# Moe

1. 下载题目所给文件。



2. 使用 binwalk 分析，只能看到一个文件，丢进 winhex，发现头和尾都是正常的，但是图片大小明显与像素不对应，于是想到会不会是两张图片，但是有一张的头部被删除了，于是搜索文本 IEND(png 对应结尾处的文本)，果然有发现：

	A4	B3	7A	CE	49	CA	F6	0F	EB	73	CC	E2	B7	81	A4	11	n°zîIEö eslâ . n°
1723376	03	EB	9F	45	47	8D	B0	59	E7	B4	1D	77	9C	9E	04	CA	eYEG ° Yç' wœž Ê
1723392	F9	4A	BB	3D	99	94	28	41	A0	BB	EA	4B	4C	A9	0D	D6	ùJ»=™( A »èKLŽ ô
1723408	55	0A	F8	F4	68	E0	0B	FF	27	06	2D	77	FD	7F	8B	05	U øôhá ý' -wý <
1723424	07	07	92	9E	6E	6D	00	00	00	00	49	45	4E	44	AE	42	' žnm IEND@B
1723440	60	82	00	00	00	0D	49	48	44	52	00	00	06	40	00	00	, IHDR @
1723456	04	B0	08	02	00	00	00	2C	63	11	C0	00	00	20	00	49	. ,c À I
1723472	44	41	54	78	01	C4	C1	4D	CC	F6	EB	7A	17	E4	E3	3C	DATx ĀĂMìøez āā<
1723488	AF	FF	7D	3F	CF	BB	DE	B5	77	CB	B6	D8	2A	69	40	4C	-ÿ)?İ»PuwÊŒ®i©L
1723504	G0	00	G0	00	G0	00	G0	00	G4	00	G0	00	G0	00	G0	00	i©L



3. 在第一个 IEND 之后，对应的格式正好是另一个 png 文件去除头部的格式，于是写入新文件，再补上头，改后缀为 png，再把第一张图片结尾后的内容删除，得到两个一个 png 文件。





4. 乍一看没什么不同，于是想到盲水印的问题，于是编写脚本，对比像素点，得到一个新的图形



5. 隐隐约约能看到有字，放在 Stegsolve 里抑或处理，得到清晰的 flag



# Easy RSA      类型: crypto

1. 下载题目所给文件，得到了一个 pem 格式的公钥和一个加密的文本，先使用 OpenSSL 提取公钥的信息。

```
C:\Users\admin\Desktop\ctf相关\rsa
λ openssl rsa -pubin -text -modulus -in publickey.pem
Public-Key: (256 bit)
Modulus:
  00:bf:e9:96:75:20:88:88:5f:2e:a2:35:2f:df:3e:
  95:15:f6:62:fc:4d:34:75:dd:a6:f8:a1:60:8e:54:
  b4:16:b7
Exponent: 65537 (0x10001)
Modulus=BFE996752088885F2EA2352FDF3E9515F662FC4D3475DDA6F8A1608E54B416B7
writing RSA key
-----BEGIN PUBLIC KEY-----
MDwwDQYJKoZIhvcNAQEBBQADKwAwKAIhAL/p1nUgiIhfLqI1L98+1RX2YvxNNHXd
pviHYI5UtBa3AgMBAAE=
-----END PUBLIC KEY-----
```

得到  $m=65537$  和  $N$  的十六进制，转成十进制得到

$N=86804467865189181998675682302645596768517985924006311724377177674474176386743$

2. 利用在线分解工具分解出

$P=293086410338424676391341741631987307899$

$Q=293086410338424676391341741631987307899$

3. 编写脚本得到密钥

```
# coding=utf-8
import math
import sys
from Crypto.PublicKey import RSA

keypair = RSA.generate(1024)

keypair.p = 293086410338424676391341741631987307899
keypair.q = 296173636181072725338746212384476813557
keypair.e = 65537

keypair.n = keypair.p * keypair.q
Qn = Long((keypair.p-1) * (keypair.q-1))

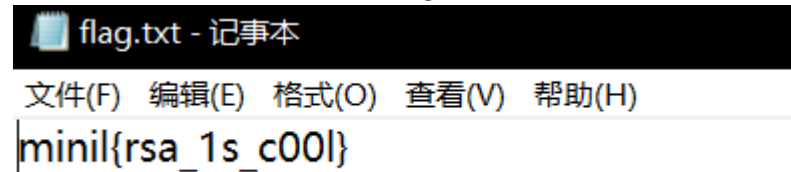
i = 1
while (True):
    x = (Qn * i) + 1
    if (x % keypair.e == 0):
        keypair.d = x / keypair.e
        break
    i += 1

private = open('private.pem', 'w')
private.write(keypair.exportKey())
private.close()
```

#### 4. 利用 OpenSSL 解密文件

```
C:\Users\admin\Desktop\ctf相关\rsa
λ openssl rsautl -decrypt -in enc1.txt -inkey private.pem -out flag.txt
```

#### 5. 打开生成的 txt 文件，得到 flag



## 我的世界      类型：MISC



轻松杀满 100 只怪，得到 flag






## PCAP 类型: MISC

1. 题目给了一个 pacpng 文件，由此判定是一道流量分析题，用 wireshark 打开这个文件

The screenshot shows the Wireshark interface with a packet capture of network traffic. The packet list on the left shows a series of DNS and NTP packets. The packet details pane on the right shows the structure of a DNS Standard query response, including the question section with PTR records for 'ipps.tcp.local' and 'ippp.tcp.local'.

2. 选择一条 tcp 对话的内容, 右键追踪 TCP 流



```

你好啊
hello. 我是你爸爸
really???
guess
I want to send you a secret
do you want it ?
sure
but it include my girlfriends name, I must keep it in a encryption one
okok...
the encryed word is:
13332zvWAL0IFG39vFS1TfgdGyITsdM30ELIESJi5BIKzTHvN5ESAkgH0GyJMdHgTf1zVM0pj1lTH1jFhELdIEIZi9XISMxz1Ary0hgrJ3A5AJD1AsIg5GxjA
rIQOvZiHsxfOGIRrvH25wF1LthG5IXImoM1MXGJLsx30gIgwAH1zJGSAUHPgyISemhznFIlPEgzGig5vHvAzhIEKIGMwxhMZH1zwlJzvIhEJZiRtIiIrF1WHGm
kZTSXISMnZg1qEhzFE1LlvJIMvGSDuw0lwyHLsH30wfhMSZIMEIKyThvWTFSESz3AGxhlXI5ytMSIJGgIkysWBHhImIg5IFgpJEizWM1MFISATHJSHI9YH1HtE1A
K1fAhIImSIWm5IIXiMGIRl3M0dELIWKZHzwEGSpHhI3HSZsLJ5jZSLvG1zJfIWKJhEFZSMqMsdTfgtEi5jEJ9LMQAwITysvSGZSMZH25rGhSq1JSGvIVJhz
TxSAIX3aiIj9sGIEPGIAqx3MGySwNjhIjySWIX0lWEGS6wRIRgJMIw25jZUAgJgzrFhAUIgMIEKARISMBF1kux0pjIKzPgGImlJytw0MAJtZ3M1MBDhhtHKlIgIvh
Xji1PLJzfEgIiIj9ZjHwZR1qIG1HE1WBMGvIFhLkZ0LFZH5Z1ZJGiMIw30AIHWIGSjZytEgWgYIMHIF03FhgMt1hFIRlPIG0vx1EiYvSiyT9ZH1InghHSFgl
HZSMRHIAOBD==
wow
what is this
the decode script will depend you intelligence~
ok let me see!
call you later~

```

得到了一段对话。

看到了一段密文，直接丢进 base64 解码，发现没有得到有价值的信息，于是看到最后 call you later 这句话，想到后面还会有对话。

3. 按照这个思路, 寻找对应 ip 地址(114.24.110.193)进行的会话, 分别为:

The image shows a Wireshark packet capture window titled "Wireshark · 追踪 TCP 流 (tcp.stream eq 1) · ITACHI.pcapng". It displays a single packet (No. 1) with a length of 78 bytes. The protocol is identified as "Application/javascript".

The raw packet data is shown in hexadecimal and ASCII format:

```
.ZC.....@...SZ..d.d.l.Z..d.d.l.Z..d.d.l.Z..d.d.l.m.Z.m.Z..d.d.Z..d.d.g.Z..d....Z..d....Z  
.d .d  
...Z  
.d....Z..d.S(...i...N(...t ...b64encodet  
...b64decodet...xxxxxxxxxxxxxxxxxxxxxxx...rot13t...b64et...caesarc...C.s"...t.t.j..d.d....}.t.j..|.|..  
S(...Nt4...ABCDEFGHGIJKLMabcdefghijklmNOPQRSTUvwxyznopqrstuvwxyz4...NOPQRSTUVWXYZnopqrstuvwxyzABCDEFGHIJKLMabcde  
fghijklm  
...t....stringt ....maketranst ....translate(...t....st...._rot13((...(s.../home/wq/ITACHI.pyR...  
...S...C...C...C...C...C...C...S  
...t..|.S(...N(...R...((...R...((...((...s.../home/wq/  
ITACHI.pyR...S...i...C...S:t.j.).)|.|.|.  
.)..t.j..|.|.|.).|.j..|.|.|.S(...N(...R...t.ascii_lowercaseR .....R  
...((...t ....plaintexttt...shiftt...alphabett...shiftd_alphabett...table((...(s.../home/wq/  
ITACHI.pyR...S...  
...C...C...S.W..d.j..t..|.....).X[t..|....D]M)..t.j..t....}.t.j..|....d...}.t....|...|....}..d.j  
..|.|...}.q".W|.S(...Ns...{2})i...s...{}{|}  
(...t...formatR...t...xranget...randomt...choicet...enc_cipherst...indext...globals(...t...ptt...cntt...tmpt...  
.ct...it...tmp(...(s.../home/wq/ITACHI.pyT)...encode...s...  
((...R...t...sysR...t...base64R...R...t...FLAGR...R...R...R...R...((...((...s.../home/wq/  
ITACHI.pyT...<module>...s... ..
```



4. 把图 4 的十六进制用 winhex 写入文件，发现对应的 ASCII 正好是图 2 对应的乱码，再结合对话中的提示，想到是一个 pyc 文件，于是直接改后缀为 pyc，丢到在线反编译工具(<http://tools.bugscaner.com/decompile/>)里，结果显示不是 pyc 文件，无法反编译。
5. 这时候回去看题面，有这么一句话：maybe 192 is telling a lie the forth time，猜测到不是 pyc 文件，于是尝试 pyo，改后缀为 pyo，再丢进工具里成功反编译，把代码保存下来：

```
import sys
import random
from base64 import b64encode, b64decode
FLAG = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXX'
enc_ciphers = [
    'rot13', 'b64e', 'caesar']

def rot13(s):
    _rot13 = string.maketrans('ABCDEFGHIJKLMabcdefghijklmNOPQRSTUVWXYZ')
    return string.translate(s, _rot13)

def b64e(s):
    return b64encode(s)

def caesar(plaintext, shift=3):
    alphabet = string.ascii_lowercase
    shifted_alphabet = alphabet[shift:] + alphabet[:shift]
    table = string.maketrans(alphabet, shifted_alphabet)
    return plaintext.translate(table)

def encode(pt, cnt):
    tmp = '2{}'.format(b64encode(pt))
    for cnt in xrange(cnt):
        c = random.choice(enc_ciphers)
        i = enc_ciphers.index(c) + 1
        _tmp = globals()[c](tmp)
        tmp = '{}{}'.format(i, _tmp)

    return tmp
```

6. 看出之前是一个加密过程，于是写出对应 decode 代码：

```
def decode(crypto):
    target = crypto
    while True:
        try:
            index = int(target[0])
            if index == 1:
                target = rot13(target[1:])
            elif index == 2:
                target = b64d(target[1:])
            elif index == 3:
                target = caesard(target[1:])
        except:
            print target
            break
```

```
def b64d(s):
    return b64decode(s)
```

```
def caesard(plaintext, shift=23):
    alphabet = string.ascii_lowercase
    shifted_alphabet = alphabet[(23):] + alphabet[:23]
    table = string.maketrans(alphabet, shifted_alphabet)
    return plaintext.translate(table)
```

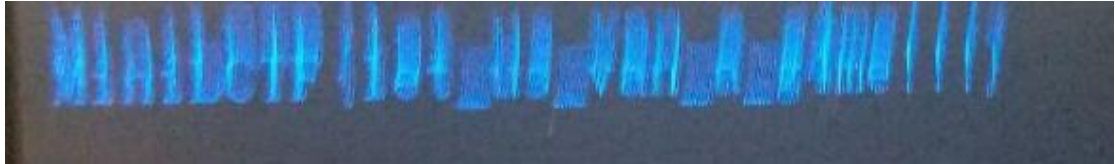
```
if __name__ == '__main__':
    encoded_flag = '13332ZvWAL0IfGJ'
    decode(encoded_flag)
```

解密得到 flag

```
λ python2 3.py
MiniLctf{1TaCH1_so_Cu7e}
```

# 贪吃蛇      类型：RE

怒赞一波队友，轻松玩到 30 分，得到 flag，



一激动拍虚了，但是能辨认出 flag：

MiniLCTF{1et\_us\_van\_a\_g4me}