蓝鲸CTF——隐写术

转载

weixin_34268753 于 2018-08-12 10:20:00 发布 1009 收藏 3
文章标签： python
原文链接：https://yq.aliyun.com/articles/649040
版权
这一次我主要是打算把蓝鲸CTF里面的有关于隐写术的题目进行一下整合，让各位能够更好的查看

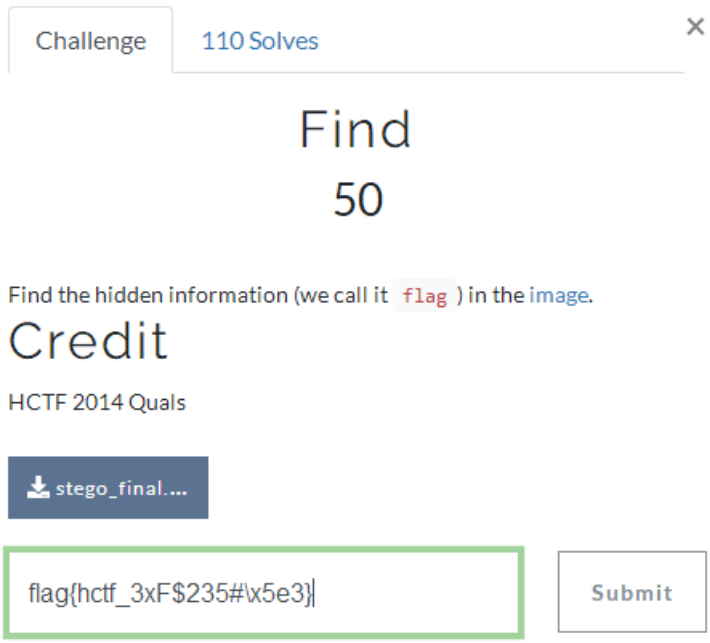
如有错误，希望各位大佬指点，谢谢！！！

1：Find

50

Find the hidden information (we call it `flag`) in the image.

HCTF 2014 Quals



解题思路：这一题难度易：

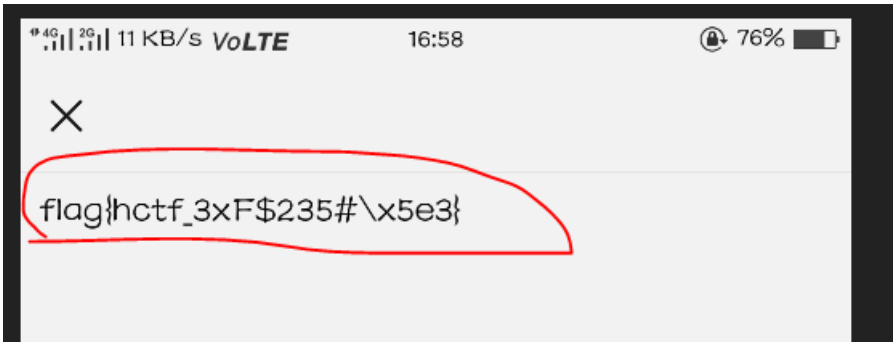
首先：下载这个图片：http://whalectf.xin/files/fc2d7e2789534ba2bdeb2bbd918990a7/stego_final.png

然后使用神器：Stegsolve（这个工具如果你们没有可以加我为好友，我会分享给你们的）

把这张照片保存就可以了（你们有没有发现这个二维码，和我们见过的不同）

使用光影魔术手这个软件，把这个照片进行反光处理就可以了

使用微信上面的扫一扫就会得到：



最后得到答案： flag{hctf_3xF$235#\x5e3}

2：Find

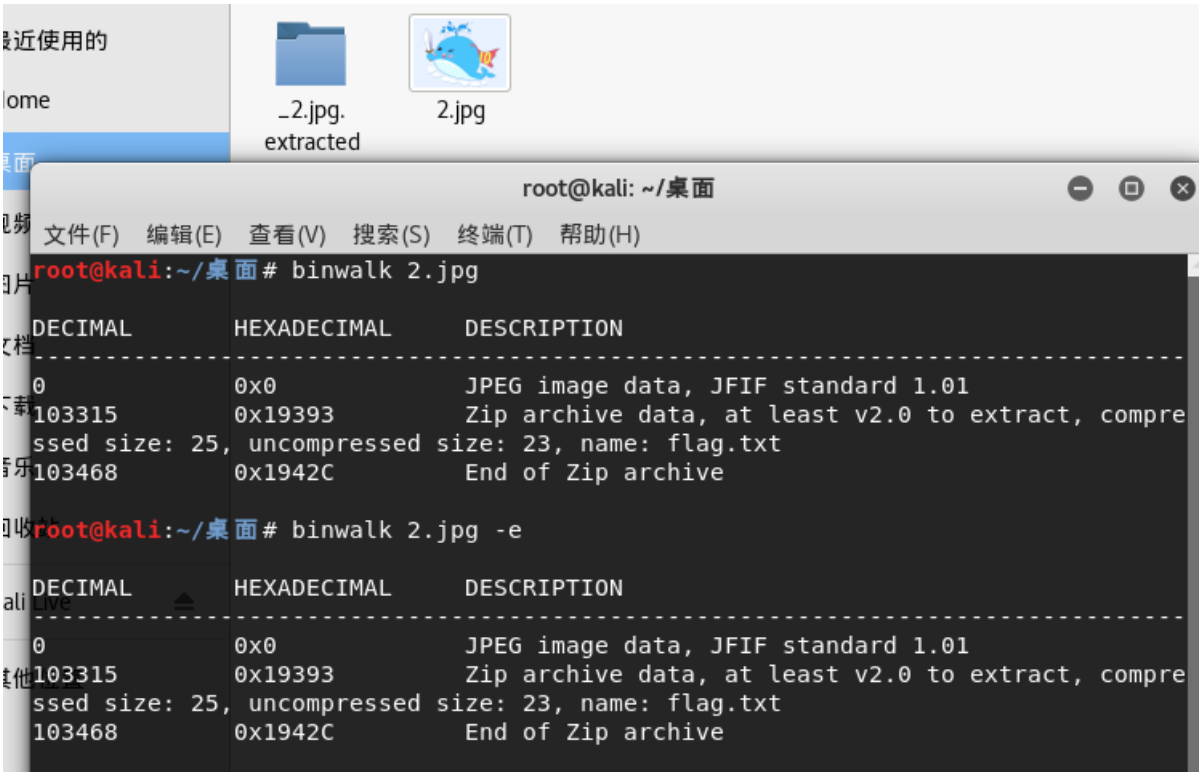
被蓝鲸吞噬的flag

whale1.jpg

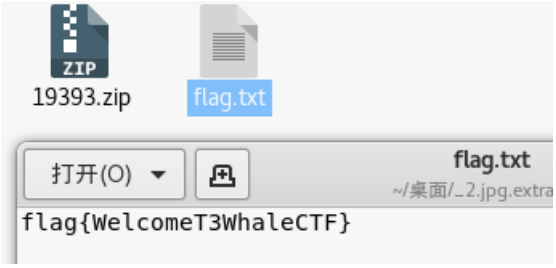
解题思路：难易程度易：

首先使用binwalk就会发现这个图片里面隐藏的有文件

然后使用binwalk xx.png -e 命令进行分离就可以得到隐藏的文件



打开文件就会得到：



最后答案：flag{WelcomeT3WhaleCTF}

3：合体鲸鱼

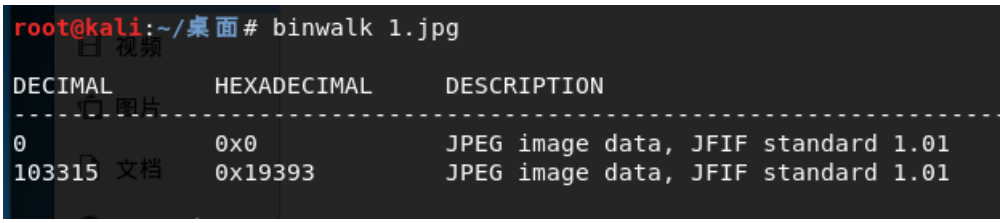
50

你能找到我不同的一面吗？

02.jpg

解题思路：难易程度易：

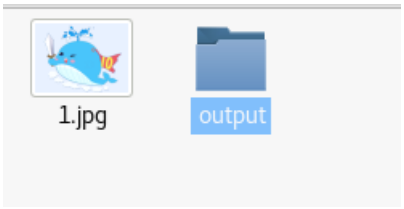
首先：这一题的思路很简单，使用binwalk就可以得到很多信息：



就会看到有隐藏的信息

然后使用foremsot xx.jpg 就会得到隐藏的文件：

```
root@kali:~/桌面# binwalk 1.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
103315        0x19393         JPEG image data, JFIF standard 1.01

root@kali:~/桌面# binwalk 1.jpg -e

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
103315        0x19393         JPEG image data, JFIF standard 1.01

root@kali:~/桌面# foremost 1.jpg
Processing: 1.jpg
|*|
root@kali:~/桌面#
```



打开就会得到：

flag{youfindmeWHALE}

最后得到答案：flag{youfindmeWHALE}

4：亚种

50

这是什么种类的蓝鲸呢？

whale.jpg

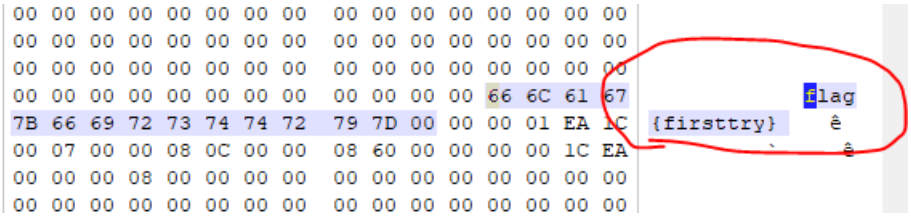
解题思路：难易程度：易

首先使用binwalk查看，然而都是看不懂的信息：

傻傻的我以为划红线的是答案所在地

点击进去都是代码看不懂

换一种方法都是我们常用的哦！！！
下面使用hex打开就可以得到答案：



最后得到答案：flag{firsttry}

5：愤怒的小猪

100

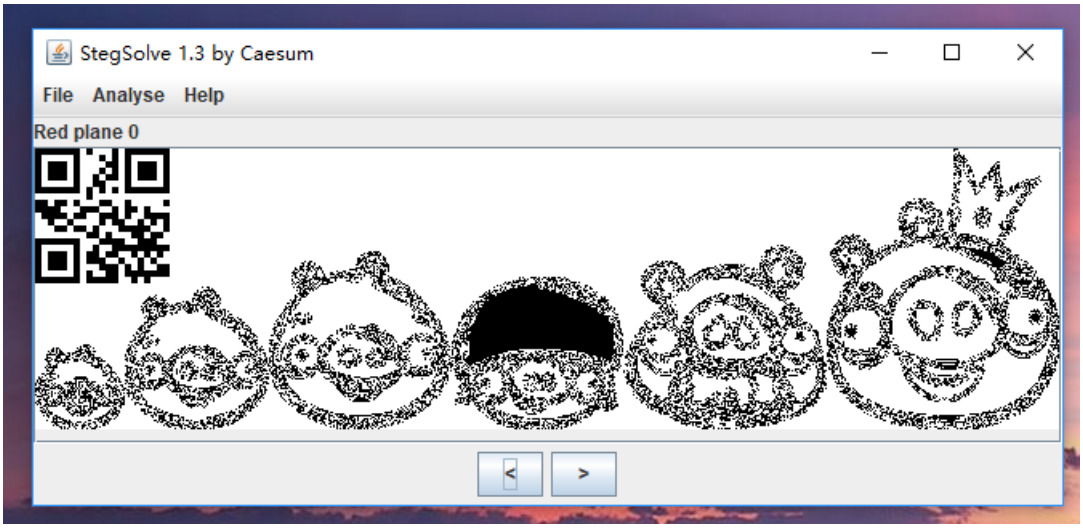
经过不断的挫折，小猪们吸取教训把宝藏藏起来了，你能找到吗？

ste.png

解题思路：难易程度易：

首先·：看见这个图片要能够想到是三原色就可以很快得到答案：

使用stegsolve就可以得到答案：



扫描上面的二维码就可以得到答案：

最后得到答案：flag{AppLeU0}

6：下雨天

50

下雨天了，你总是在我心中一闪而过

Misc01.jpg

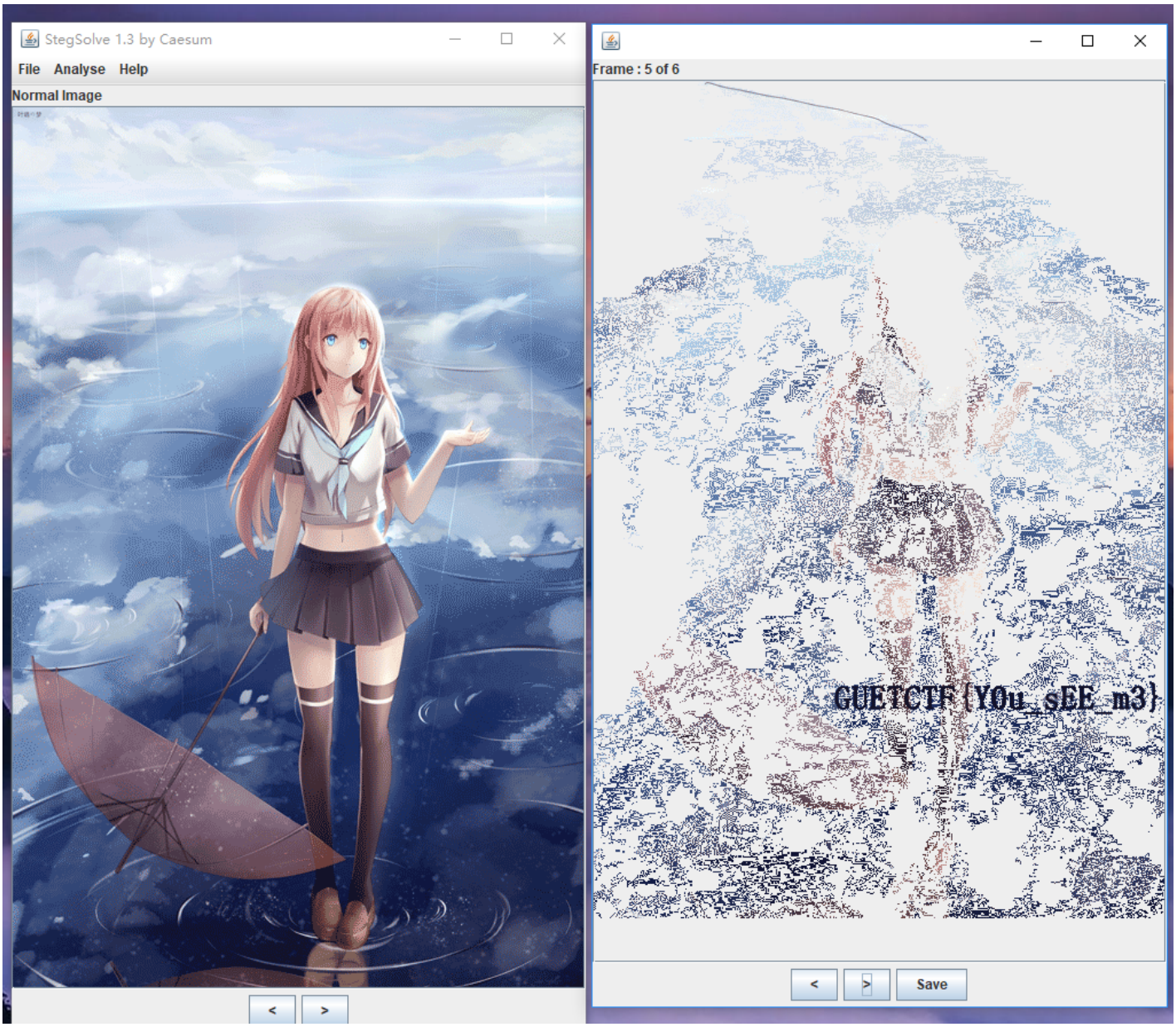
解题思路：难易程度易：

这一题我还是请教大佬之后写出来的，而我之所以写成易

是因为它真的很简单，

是我想太多了，当时没有写出来

这一题是需要使用Stegolove就可以做出来了



最后得到答案：GUETCTF{Y0u_sEE_m3}

下面是我错误的步骤，写出来是为了能够告诫自己

开始的时候看见是图片

直接想到使用binwalk就会得到答案

是的这一题分离出来一张图片

然而这个图片并没有什么有用的信息

接着使用 winhex就会发现文件头不对

更改一下文件头发现图片打不开了

（QAQ)

经历了那么多，最后发现那么简单，我只是感觉自己很笨呀（QAQ)

7：这是什么

100

兔子的脚下踩到了什么？

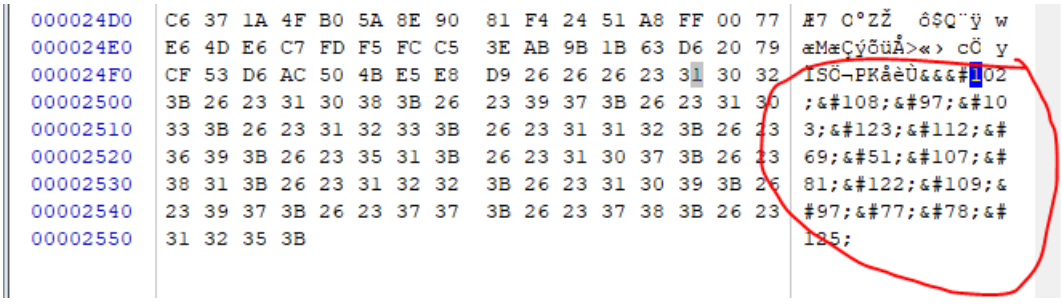
rabbit.jpg

解题思路：难易程度易：

这一题比较简单，使用winhex就可以得到最后的答案

（为什么那么简单的题目我要想的那么复杂，导致现在开始怀疑自己QAQ）

根据题目的描述就能够想到最后的结果所在地（拉到最后的地方就会得到）



这个是ASCII码，使用ASCII码表对照一下就会得到最后的答案：

ASCII表

（ American Standard Code for Information Interchange 美国标准信息交换代码 ）

高四位 / 低四位		ASCII控制字符												ASCII打印字符															
		0000						0001						0010		0011		0100		0101		0110		0111					
		0						1						2		3		4		5		6		7					
		十进制	字符	Ctrl	代码	转义字符	字符解释	十进制	字符	Ctrl	代码	转义字符	字符解释	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	Ctrl			
0000	0	0		^@	NUL	\0	空字符	16	▶	^P	DLE		数据链路转义	32		48	0	64	@	80	P	96	`	112	p				
0001	1	1	☺	^A	SOH		标题开始	17	◀	^Q	DC1		设备控制 1	33	!	49	1	65	A	81	Q	97	a	113	q				
0010	2	2	☻	^B	STX		正文开始	18	↕	^R	DC2		设备控制 2	34	"	50	2	66	B	82	R	98	b	114	r				
0011	3	3	♥	^C	ETX		正文结束	19	‼	^S	DC3		设备控制 3	35	#	51	3	67	C	83	S	99	c	115	s				
0100	4	4	♦	^D	EOT		传输结束	20	¶	^T	DC4		设备控制 4	36	$	52	4	68	D	84	T	100	d	116	t				
0101	5	5	♣	^E	ENQ		查询	21	§	^U	NAK		否定应答	37	%	53	5	69	E	85	U	101	e	117	u				
0110	6	6	♠	^F	ACK		肯定应答	22	▬	^V	SYN		同步空闲	38	&	54	6	70	F	86	V	102	f	118	v				
0111	7	7	•	^G	BEL	\a	响铃	23	↨	^W	ETB		传输块结束	39	'	55	7	71	G	87	W	103	g	119	w				
1000	8	8	◘	^H	BS	\b	退格	24	↑	^X	CAN		取消	40	(	56	8	72	H	88	X	104	h	120	x				
1001	9	9	○	^I	HT	\t	横向制表	25	↓	^Y	EM		介质结束	41	)	57	9	73	I	89	Y	105	i	121	y				
1010	A	10	◙	^J	LF	\n	换行	26	→	^Z	SUB		替代	42	*	58	:	74	J	90	Z	106	j	122	z				
1011	B	11	♂	^K	VT	\v	纵向制表	27	←	^[	ESC	\e	溢出	43	+	59	;	75	K	91	[	107	k	123	{				
1100	C	12	♀	^L	FF	\f	换页	28	∟	^\	FS		文件分隔符	44	,	60	<	76	L	92	\	108	l	124	|				
1101	D	13	♪	^M	CR	\r	回车	29	↔	^]	GS		组分隔符	45	-	61	=	77	M	93	]	109	m	125	}				
1110	E	14	♫	^N	SO		移出	30	▲	^^	RS		记录分隔符	46	.	62	>	78	N	94	^	110	n	126	~				
1111	F	15	☼	^O	SI		移入	31	▼	^-	US		单元分隔符	47	/	63	?	79	O	95	_	111	o	127	⌂	^Backspace 代码：DEL			

注：表中的ASCII字符可以用“Alt + 小键盘上的数字键”方法输入。 2013/08/08

最后得到答案：flag{pE3kQzmaMN}

8：真是动图

100

真的是动图，相信我啊，我就把password藏里面了
答案格式key{xxx}

gif.gif

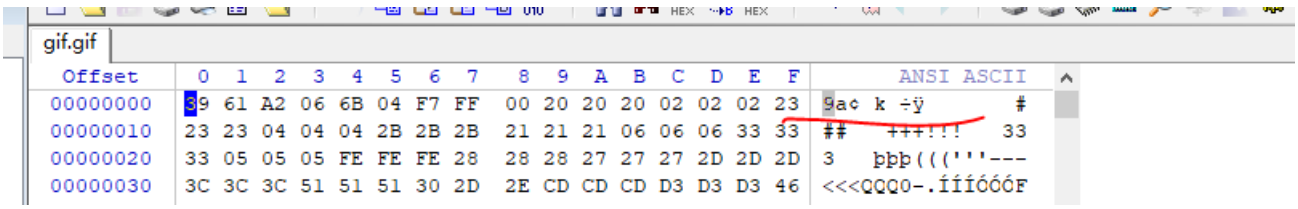
解题思路：难易程度：中等（之所以评为中等，并不是因为这一题难，而是因为不容易读

至少我是这样认为的QAQ)

首先：需要下载图片，使用百度云，360浏览器，以及所有可以使用链接下载的工具都可以了：

然后：就会发现图片打开出错

放进winhex中看一下就会发现，文件头缺失

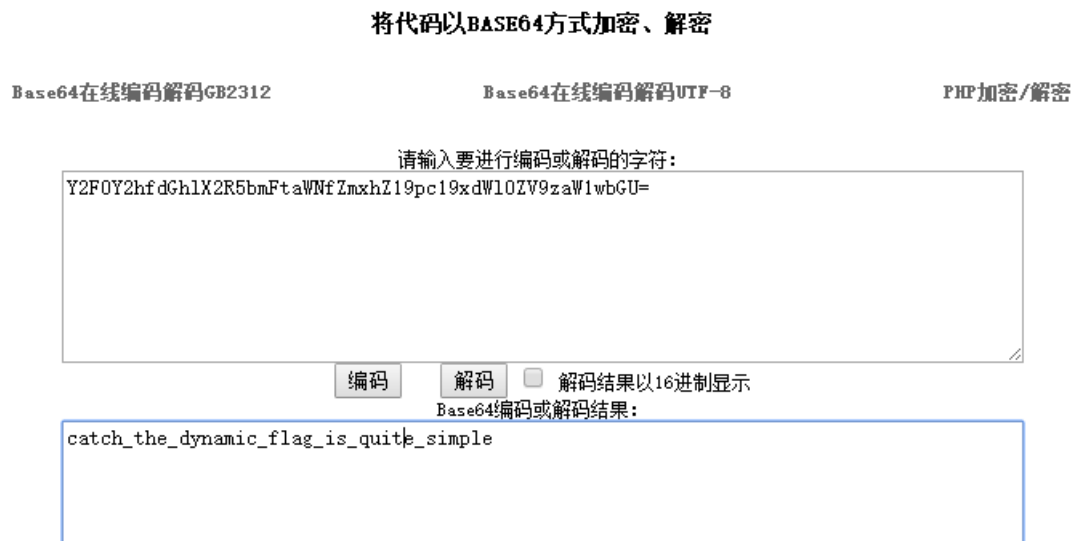


加上GIF8就会保存之后就会得到：

这是一个动图：使用steglove打开发现，看的不清楚，使用2345看图王吧
毕竟这个可以一帧一帧的看

把灰色的部分读取就会得到base64编码的东西：

使用在线工具：



就会得到最后的答案：

key{catch_the_dynamic_flag_is_quite_simple}

9：错误压缩

150

把我的女神压缩就忍了，竟然还压缩错了！
答案格式：SCTF{xxx}

sctf.png

解题思路：总的来说这一题，还是比较难的，可能是自己会的太少，所以这样感觉的

不过这一题还是会学习到许多新的知识的（HHH)。

首先：我们会使用 binwalk 看一下有没有隐藏的文件或者压缩包（但是都没有QAQ)
使用stegolve 也没有任何信息（QAQ)

然后：上网搜一下看看有没有大佬写过这个题目的wp（HHH很聪明呀！！没有什么都没有QAQ)

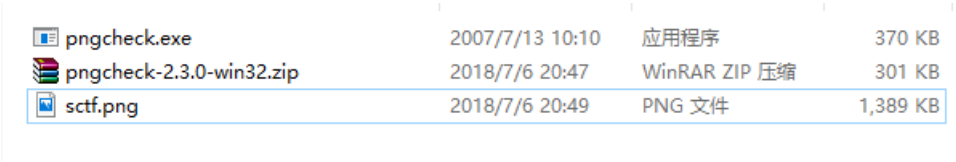
自己解决吧（自己又上网搜一些关于图片隐写的解题思路，发现pngcheck没有使用过）

使用一一下吧（但是没有工具，自己又蒙蒙的上网找工具（http://www.libpng.org/pub/png/apps/pngcheck.html)

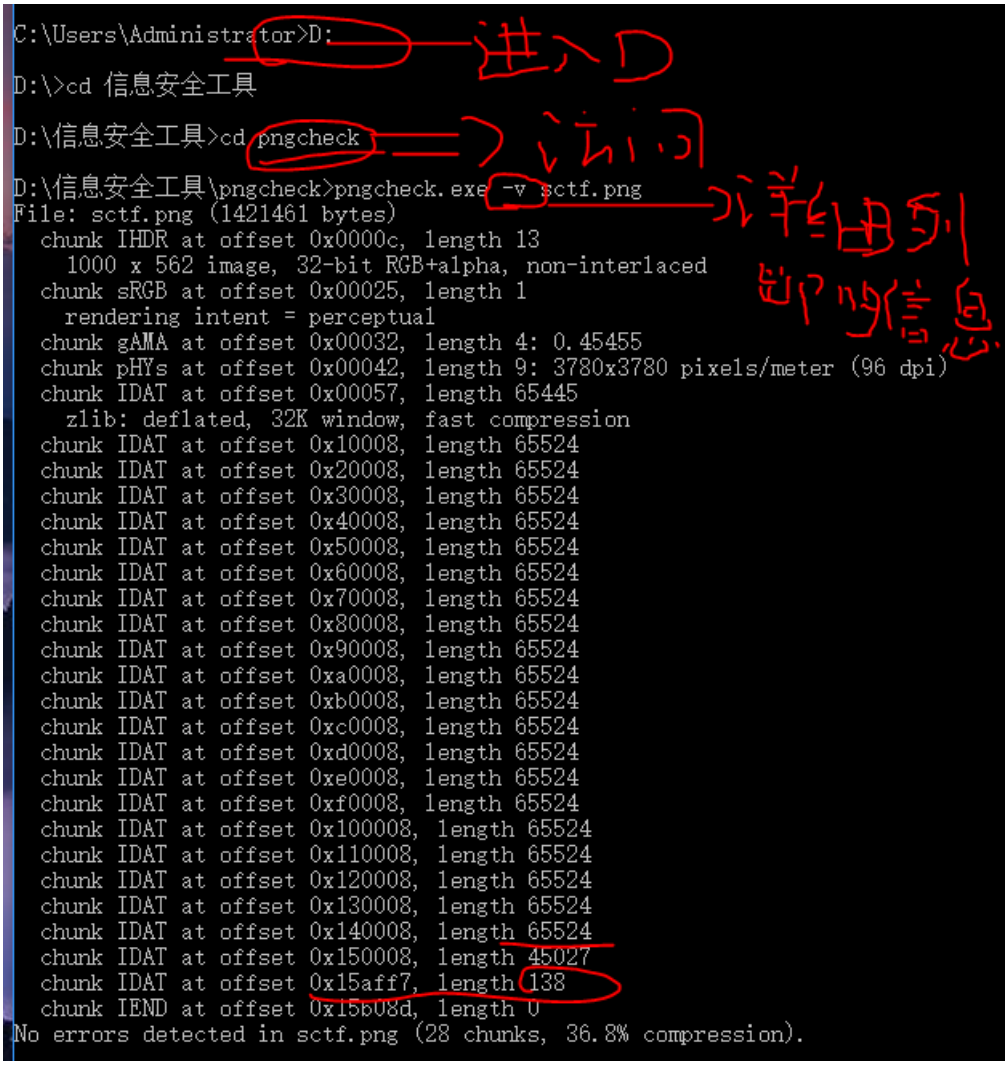
这个工具的使用方法比较简单（具体使用方法，我会在下面解题思路中详细写出来的）

使用cmd（这个cmd今天不知道怎么了就是进不了文件夹里面的内容了，找到一位大佬用另一种方法进入的（QAQ))

我们需要把图片和pngcheck放在同一个文件夹里面



然后使用cmd：

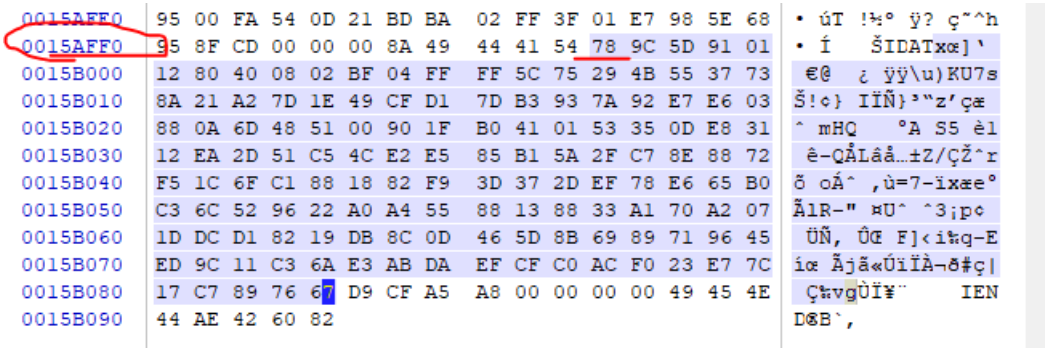


可以看到，正常的块的length是在65524的时候就满了，而倒数第二个IDAT块长度是45027，

最后一个长度是138，很明显最后一个IDAT块是有问题的，因为他本来应该并入到倒数第二个未满的块里

看到下面的数据和上面的不同，关键点就是在这个地方

使用winhex找到（如下图所示的红线的位置）



查找78 9C文件头标志，这个地方需要使用python进行处理就会得到：

```
import zlib
import binascii
IDAT =("789C5D91011280400802BF04FFFF5C75294B5537738A21A27D1E49CFD17D
B3937A92E7E603880A6D485100901FB0410153350DE83112EA2D51C54CE2E585B15A
2FC78E8872F51C6FC1881882F93D372DEF78E665B0C36C529622A0A45588138833A1
70A2071DDCD18219DB8C0D465D8B6989719645ED9C11C36AE3ABDAEFCFC0ACF023E77
C17C7897667").decode("hex") result = binascii.hexlify(zlib.decompress(IDAT)) print result
```

就会得到：

使用conventer进行hex转化就会得到：

如果ctf做的比较多就会发现，关于0101:

（1）有可能代表的是ASCII码

（2）有可能是二维码：0表示的是黑色，1表示的是白色

关于这个里面的0101，一共有625个，正好是25的平方

借鉴一段代码：

```
import Image
MAX = 25
pic = Image.new("RGB",(MAX,MAX)) str =
"11111110001000011011111111000001011100101101000001101110101000000000101110110
11101001000000001011101101110101110110100101110110000010101011011010000011111111010
10101011111111000000001011101110000000011010011000001010011101101111010101010010000111
00000000000101000000001001001101000100111001111011100111100000111011111100011001010001
10011100001010100011010001111010110000010100010110000011011101100100001110011100100
01011111111010000000011010100100011110111111110111000011010110111000000100001100100011
11010111010001101001111100001011101011000111010011100101110100100111011011000100000
10110001101000110001111111011010110111011011" i=0 for y in range (0,MAX): for x in range (0,MAX): if(str[i]
== '1'): pic.putpixel([x,y],(0,0,0)) else: pic.putpixel([x,y],(255,255,255)) i = i+1 pic.show()
pic.save("flag.png")
```

这个需要安装Image库

运行之后就会得到：

使用手机或者在线工具都可以，但是我使用的是QR Research

就会得到答案：

注意flag的格式，以及需要看清0和o的区别，1和l的不同

10：模糊的图片

100

p图大神请帮我把模糊的图片弄清楚，答案格式：flag{xxxx}

1.png

Challenge 13 Solves

模糊的图片

100

p图大神请帮我把模糊的图片弄清楚，答案格式：flag{xxxx}

1.png

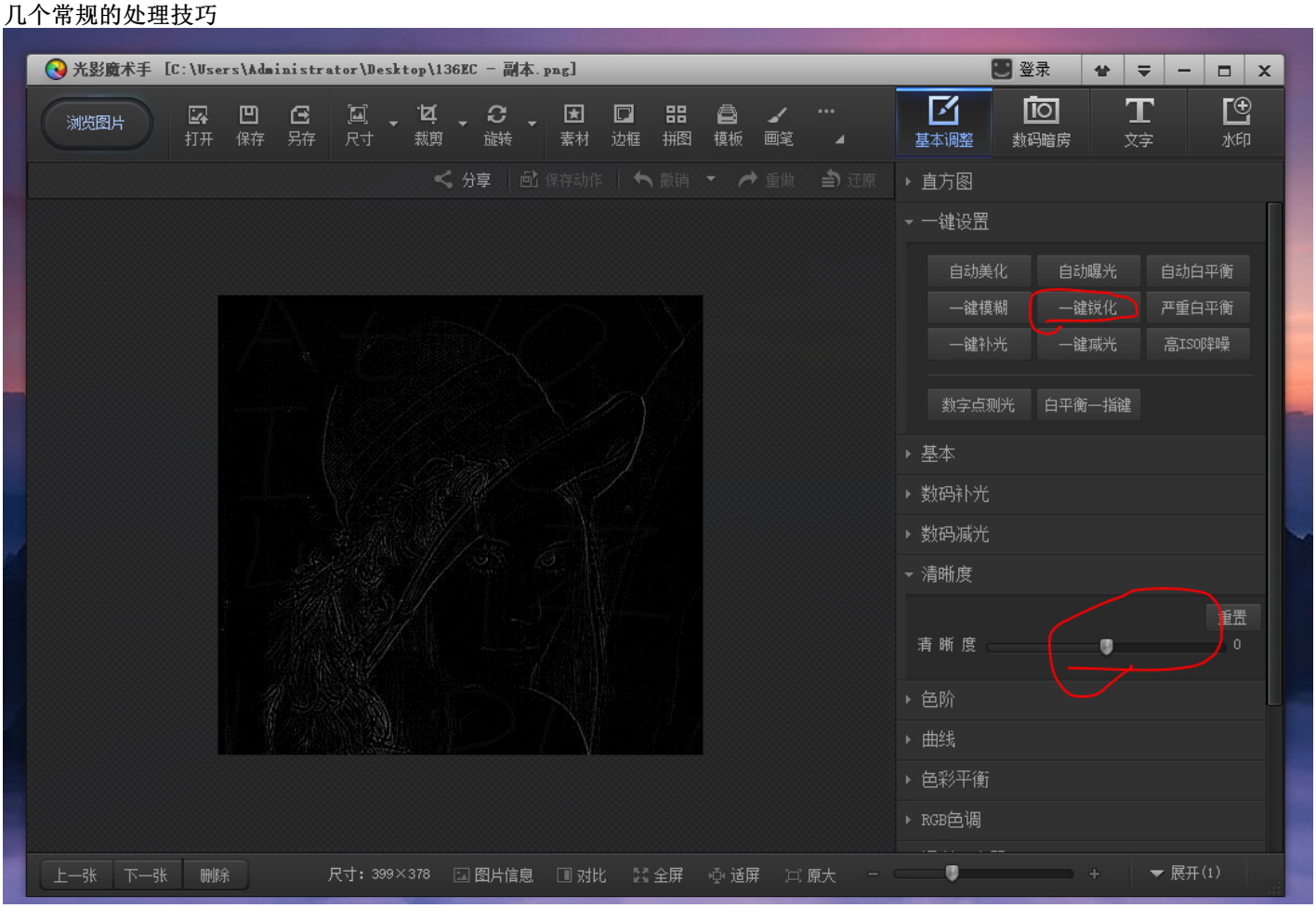
解题思路：关于这一题有两种解题思路
但是总的来说是一种都是使用一个思想锐化的思维

可能是我太弱了

导致这一题并没有得到最后的答案

首先：看到题目
”p图大神请帮我把模糊的图片弄清楚“
这一句已经很清楚的知道需要使用ps以及其它图片处理工具进行锐化处理

这个地方可以使用图片处理工具有：美图秀秀，ps，光影魔术手
（我比较推荐使用光影魔术手，没有广告）
几个常规的处理技巧



经过处理之后就会得到：

或者使用python：

```
#coding:utf-8

import Image

img = Image.open('ifs.bmp') X = img.size[0] Y = img.size[1] print X,Y for i in range(X-2): for j in range(Y-2): a = img.getpixel((i,j))[0]+img.getpixel((i,j))[1]+img.getpixel((i,j))[2] b = img.getpixel((i,j+1))[0]+img.getpixel((i,j+1))[1]+img.getpixel((i,j+1))[2] c = img.getpixel((i,j+2))[0]+img.getpixel((i,j+2))[1]+img.getpixel((i,j+2))[2] if (a > b and c > b) or (a < b and c < b): pass else: img.putpixel((i,j),(255,255,255)) img.show()
```

看图得key: At10ISCC421ZLAPL

11：IHDR

100

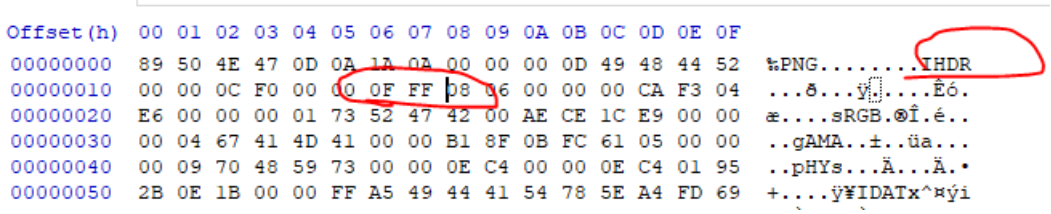
苹果拍照很美，但是我要拍的东西怎么老是拍不出来

HARD.png

解题思路：难易程度易：

直接使用winhex就可以得到答案：
首先：使用winhex打开找到IHDR位置就可以了：

然后知道更改高度就是划红线的地方：



保存之后就会得到最后的答案：FLAG{ihDR_ALSO_FUN}

12:斗鸡眼

200

听说斗鸡眼就能找到答案了
答案格式：ISG{xxxx}

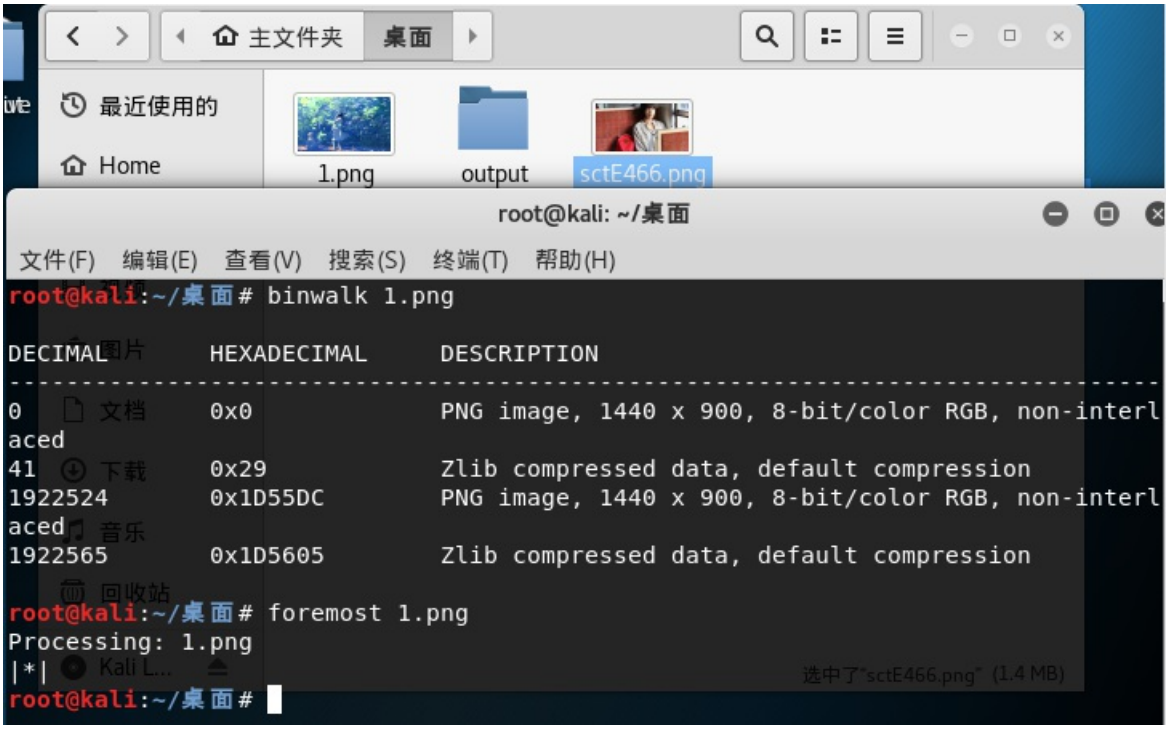
final.png

解题思路：难易程度难：

首先这一题虽然是实验吧的原题，但是当时的我并没有做出来，这个是我的问题（毕竟哪个时候能力不够呀）

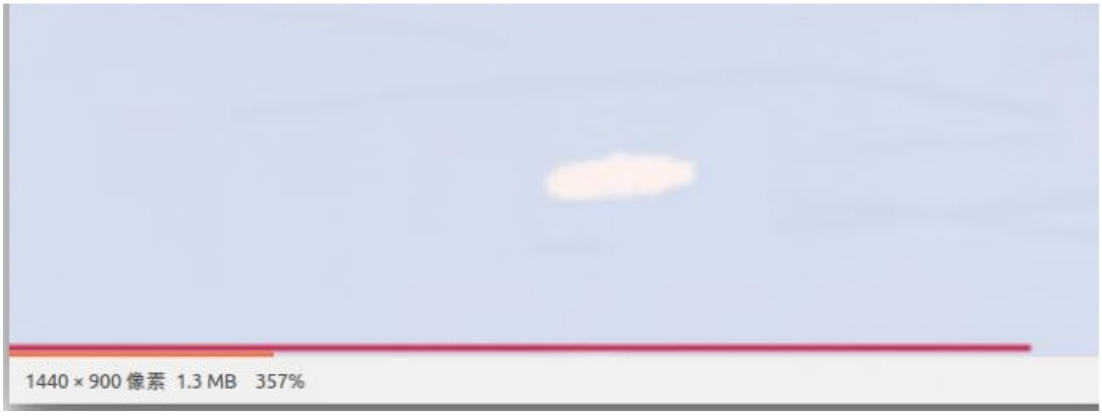
虽然今天写出来了解题思路，但是还是参考一些大佬写的思路，自己做的总结（太弱了，是我现在的问题）

首先:关于这一题使用binwalk



以及foremost就会得到一个新的文件，是两个看起来相同的照片，然而并没有那么简单的哦！！

是的这是两张不同的图片，用Beyond Compare 在二进制下看一眼，文件的最末尾果然不同



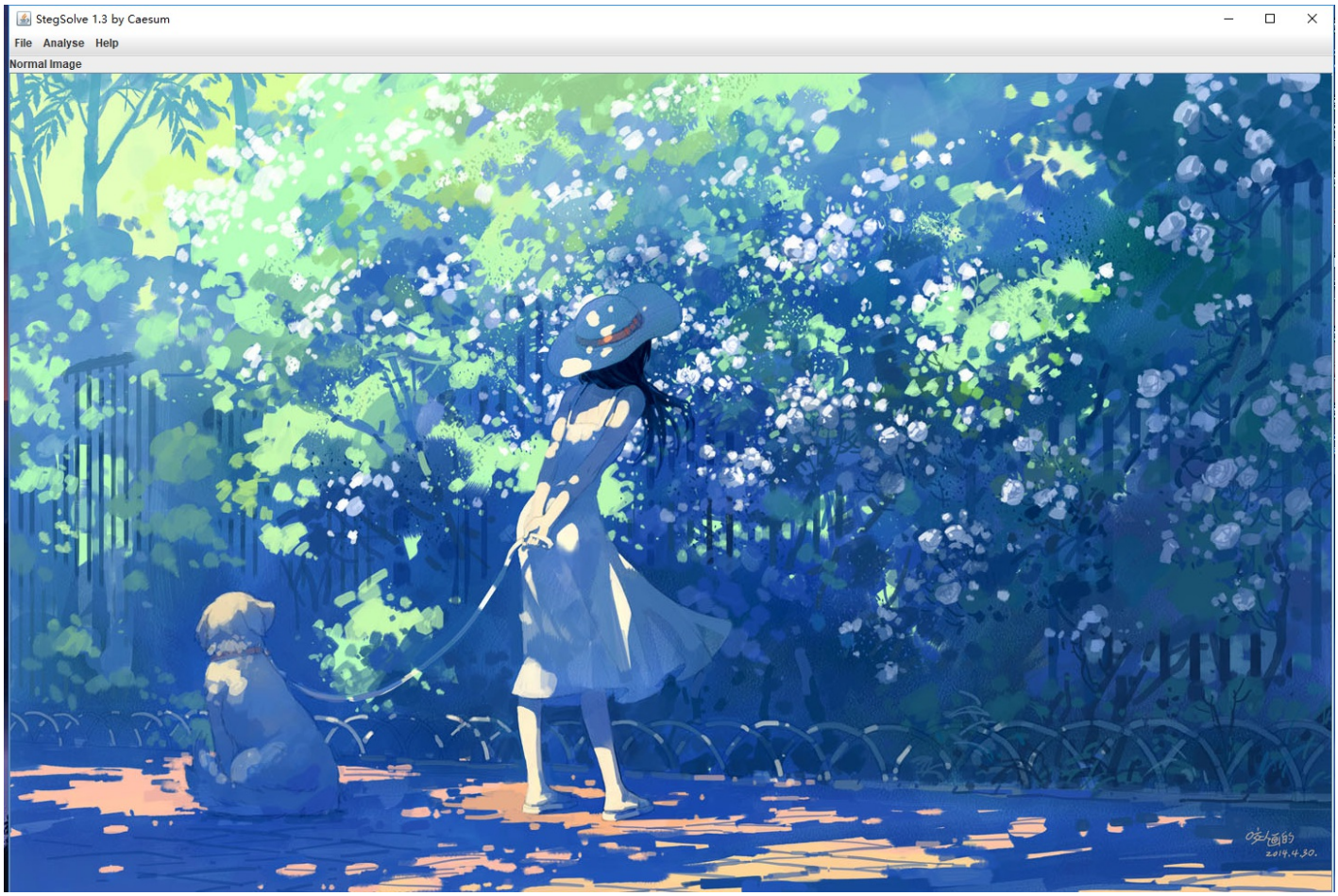
竟然看出了什么∑(っ °Д °;)っ 这一条红线是啥玩意？（・(エ)・）

00000000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	‰PNG IHDR
00000016	00	00	05	A0	00	00	03	84	08	02	00	00	00	D8	2F	01	„ Ø/
00000032	85	00	01	00	00	49	44	41	54	78	9C	BC	FD	57	B3	6C	… IDATxœ¼ýW³l
00000048	4B	92	1E	88	B9	88	88	25	52	6D	71	F4	B9	AA	EA	56	K’ ^¹^^%Rmqô¹ªêV
00000064	75	55	0B	34	80	86	26	7B	0C	63	E0	D0	86	0F	63	9C	uU 4€†&{ càÐ† cœ
00000080	67	3E	F0	F7	F1	85	C6	17	92	83	01	8D	30	62	80	69	g>ð÷ñ…Æ ’ƒ 0b€i
00000096	88	6E	4C	8B	EA	AE	EA	52	57	1F	BD	65	8A	A5	22	C2	ˆnL‹ê®êRW ½eŠ¥"Â
00000112	DD	F9	B0	32	73	E7	D9	67	9F	DB	55	40	01	61	D7	CE	Ýù°2sçÙgŸÛU@ a×Î
00000128	CD	BD	73	ED	B5	62	79	78	78	B8	7F	AE	F0	3C	FF	CB	Í½síµbyxx¸ ®ð<ÿË
00000144	9E	A0	B1	B6	A4	E0	81	04	F4	E7	AF	EF	FD	BF	FE	E7	ž ±¶¤à ôç¯ïý¿þç
00000160	0A	EB	C1	15	0E	20	E7	1E	2B	B8	6F	7D	F3	F8	E1	B3	ëÁ ç +¸o}óøá³
00000176	FF	F1	BF	7F	88	70	86	E0	B5	B1	50	40	76	EE	4F	9E	ÿñ¿ ˆp†àµ±P@vîOž
00000192	6D	FE	D5	4F	E0	D5	F2	D1	FF	F5	1F	7F	F1	9D	C7	F7	mþÕOàÕòÑÿõ ñ Ç÷
00000208	EF	BB	42	F4	32	C7	A6	28	BD	01	35	32	F4	38	9D	91	ï»Bô2Ç¦(½ 52ô8 ‘
00000224	39	98	FC	C5	8B	FC	7F	FF	57	2F	EB	7B	9F	7E	F2	D1	9˜üÅ‹ü ÿW/ë{Ÿ~òÑ
00000240	A2	5F	FD	FC	12	16	69	33	6F	56	31	14	FD	D0	4B	B3	¢_ýü i3oV1 ýÐK³
00000256	2E	C2	A4	CA	78	EE	30	98	19	EC	C6	FE	F3	FF	F8	7B	.Â¤Êxî0˜ ìÆþóÿø{
00000272	6F	BE	FF	DD	A7	20	D7	64	ED	AC	9A	B4	AA	EB	0D	B4	o¾ÿÝ§ ×dí¬š´ªë ´
00000288	03	E4	04	5C	54	4D	84	CB	0D	6E	FA	70	BD	C4	D5	CA	ä \TM„Ë núp½ÄÕÊ
00000304	44	4B	87	E8	9C	43	44	11	49	29	31	73	5D	D7	75	5D	DK‡èœCD I)1s]×u]
00000320	9F	F0	E6	C1	A3	C7	17	CB	E5	67	5F	3F	CF	40	E6	7D	Ÿðæ£Ç Ëåg_?Ï@æ}
00000336	9F	8D	9D	7B	B3	EE	75	37	CC	0C	11	89	08	11	CD	4C	Ÿ {³îu7Ì ‰ ÍL
00000352	72	F4	8C	93	B2	AC	82	77	60	84	E6	88	01	07	97	CB	rôŒ“²¬‚w`„æˆ —Ë
00000368	49	39	6F	34	2D	D3	30	0D	F0	BB	4F	D3	1F	FE	56	45	I9o4-Ó0 ð»OÓ þVE
00000384	B0	4E	70	1D	20	01	B8	3E	F9	00	75	E1	2D	A6	8B	D2	°Np ¸>ù uá-¦‹Ò
00000400	FF	F0	E7	E7	F1	CF	7E	21	67	FD	B4	15	E8	E3	72	5E	ÿðççñÏ~!gý´ èãr^
00000416	50	E9	E8	A2	EB	10	59	93	12	70	E1	0B	0F	4E	15	1C	Péè¢ë Y“ pá N
00000432	52	B2	8D	21	80	39	45	CC	4A	49	20	67	55	43	8B	03	R² !€9EÌJI gUC‹
00000448	12	A1	63	31	8D	92	53	CE	0A	66	84	73	0D	C4	10	02	¡c1 ’SÎ f„s Ä
00000464	87	82	44	FB	94	3B	66	A8	27	65	7D	14	48	AF	B3	75	‡‚Dû”;f¨'e} H¯³u
00000480	03	9C	40	F9	71	6A	57	F7	E9	CB	3F	FC	BD	FA	C3	EA	œ@ùqjW÷éË?ü½úÃê
00000496	B5	42	00	28	32	90	82	19	28	01	33	F0	22	36	4C	44	µB (2 , (3ð"6LD
00000512	0E	23	D0	A5	F9	5F	5E	84	3F	FF	CA	7F	F1	02	AF	DD	#Ð¥ù_^„?ÿÊ ñ ¯Ý
00000528	83	12	22	6E	5E	7E	7C	B4	FC	6F	FF	EE	FC	D3	7B	E2	ƒ "n^~|´üoÿîüÓ{â
00000544	F5	BA	24	FC	F3	CD	C3	BF	FC	AB	D5	57	2F	CA	01	8E	õº$üóÍÃ¿ü«ÕW/Ê Ž
00000560	06	8C	E8	06	E7	29	45	5B	97	A7	1A	3B	9F	DB	05	F7	Œè ç)E[—§ ;ŸÛ ÷

使用winhex查看一下，看看里面有没有藏什么东西(￣.￣)，小样你躲不过我专职酱油选手的，

然而，并没有发现什么重要的信息（QAQ)

但是，难不倒本宝宝的，我还有神器steglove工具进行合并：



合并一下并且保存：默认solved.bmp

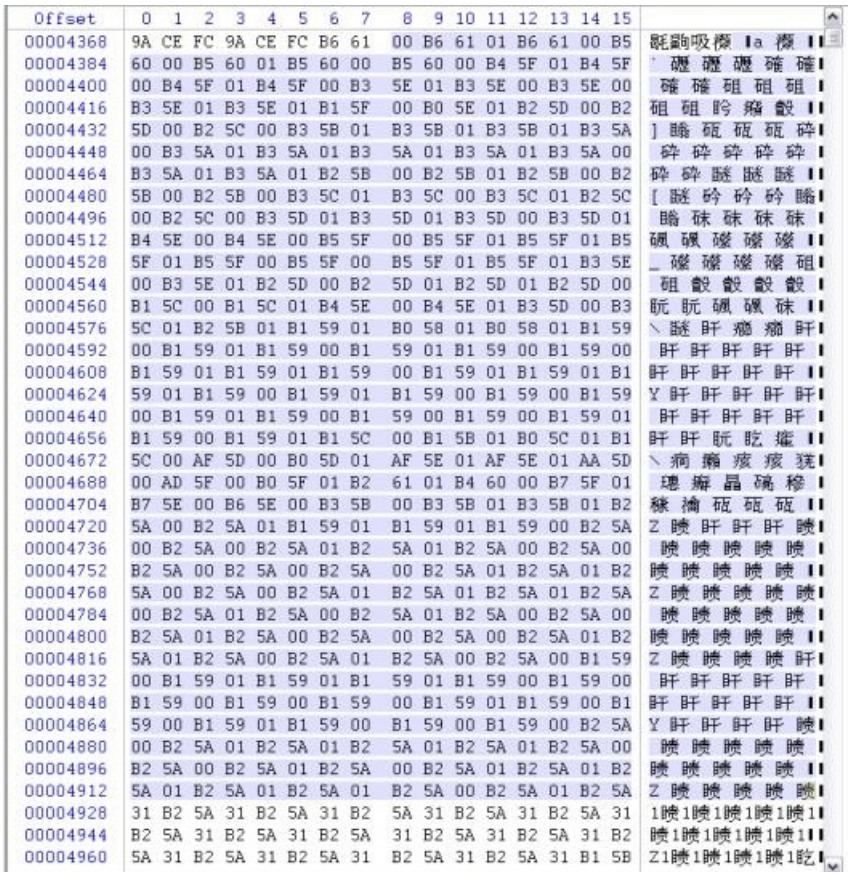
使用winhex进行查看就会得到：

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ANSI ASCII
00000000	42	4D	B6	53	3B	00	00	00	00	00	36	00	00	00	28	00	BM¶S; 6 (
00000016	00	00	A0	05	00	00	84	03	00	00	01	00	18	00	00	00	„
00000032	00	00	80	53	3B	00	00	00	00	00	00	00	00	00	00	00	€S;
00000048	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000064	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000096	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000112	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000128	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000144	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000176	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000192	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000208	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000224	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000240	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000256	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000272	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000288	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000304	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000320	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000336	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000352	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000368	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000384	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

记住千万不要使用进度条乱拉，这样会错过重要的信息：

00004320	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00004336	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00004352	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00004368	00	00	00	00	00	00	00	00	34	00	00	35	00	00	34	00	4 5 4
00004384	00	33	00	00	32	00	00	33	00	00	33	00	00	33	00	00	3 2 3 3 3
00004400	32	00	00	33	00	00	32	00	00	30	00	00	31	00	00	31	2 3 2 0 1 1
00004416	00	00	30	00	00	30	00	00	30	00	00	2E	00	00	2F	00	0 0 0 . /
00004432	00	2F	00	00	2E	00	00	2C	00	00	2C	00	00	2C	00	00	/ . , , ,
00004448	2E	00	00	2F	00	00	2F	00	00	2F	00	00	2F	00	00	2E	. / / / / .
00004464	00	00	2F	00	00	2F	00	00	2F	00	00	2E	00	00	2F	00	/ / / . /
00004480	00	2F	00	00	2F	00	00	31	00	00	30	00	00	31	00	00	/ / 1 0 1
00004496	2E	00	00	2E	00	00	2E	00	00	2E	00	00	2F	00	00	2E	. . . . / .
00004512	00	00	30	00	00	30	00	00	31	00	00	30	00	00	30	00	0 0 1 0 0
00004528	00	30	00	00	31	00	00	31	00	00	30	00	00	30	00	00	0 1 1 0 0
00004544	31	00	00	30	00	00	30	00	00	31	00	00	31	00	00	30	1 0 0 1 1 0
00004560	00	00	2F	00	00	2E	00	00	30	00	00	31	00	00	2F	00	/ . 0 1 /
00004576	00	31	00	00	2E	00	00	31	00	00	2E	00	00	31	00	00	1 . 1 . 1
00004592	30	00	00	31	00	00	30	00	00	31	00	00	30	00	00	30	0 1 0 1 0 0
00004608	00	00	31	00	00	31	00	00	30	00	00	31	00	00	31	00	1 1 0 1 1
00004624	00	31	00	00	30	00	00	31	00	00	30	00	00	30	00	00	1 0 1 0 0
00004640	30	00	00	31	00	00	30	00	00	30	00	00	30	00	00	31	0 1 0 0 0 1
00004656	00	00	30	00	00	31	00	00	2F	00	00	30	00	00	33	00	0 1 / 0 3
00004672	00	34	00	00	34	00	00	36	00	00	39	00	00	39	00	00	4 4 6 9 9
00004688	3C	00	00	3B	00	00	3D	00	00	3D	00	00	3C	00	00	3B	< ; = = < ;
00004704	00	00	39	00	00	36	00	00	32	00	00	33	00	00	33	00	9 6 2 3 3
00004720	00	31	00	00	30	00	00	31	00	00	31	00	00	30	00	00	1 0 1 1 0
00004736	31	00	00	31	00	00	30	00	00	30	00	00	31	00	00	31	1 1 0 0 1 1
00004752	00	00	31	00	00	31	00	00	31	00	00	30	00	00	30	00	1 1 1 0 0
00004768	00	31	00	00	31	00	00	30	00	00	30	00	00	30	00	00	1 1 0 0 0
00004784	31	00	00	30	00	00	31	00	00	30	00	00	31	00	00	31	1 0 1 0 1 1
00004800	00	00	30	00	00	31	00	00	31	00	00	31	00	00	30	00	0 1 1 1 0
00004816	00	30	00	00	31	00	00	30	00	00	31	00	00	31	00	00	0 1 0 1 1
00004832	30	00	00	31	00	00	31	00	00	31	00	00	30	00	00	30	0 1 1 1 0 0
00004848	00	00	30	00	00	30	00	00	30	00	00	31	00	00	30	00	0 0 0 1 0
00004864	00	30	00	00	31	00	00	30	00	00	30	00	00	30	00	00	0 1 0 0 0
00004880	31	00	00	30	00	00	30	00	00	30	00	00	30	00	00	31	1 0 0 0 0 1
00004896	00	00	31	00	00	30	00	00	31	00	00	30	00	00	30	00	1 0 1 0 0
00004912	00	30	00	00	30	00	00	30	00	00	31	00	00	30	00	00	0 0 0 1 0
00004928	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

就在这里，拉进度条的小伙伴你是看不到的
咳咳，这里其实很难进行下一步，因为不知道要干嘛了，把这一块抠出来，然后各种编码转换
然并卵o(￣ヘ￣*o)
后来我发现了，这估计就是一个偏移的提示
Tips：把start.png和to.png保存为bmp格式，因为png存储的问题，不太好找偏移
然后瞎折腾啊！！！！！！
终于发现在to.bmp的对应偏移里找到了些东西



Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
00004368	9A	CE	FC	9A	CE	FC	B6	61	00	B6	61	01	B6	61	00	B5
00004384	60	00	B5	60	01	B5	60	00	B5	60	00	B4	5F	01	B4	5F
00004400	00	B4	5F	01	B4	5F	00	B3	5E	01	B3	5E	00	B3	5E	00
00004416	B3	5E	01	B3	5E	01	B1	5F	00	B0	5E	01	B2	5D	00	B2
00004432	5D	00	B2	5C	00	B3	5B	01	B3	5B	01	B3	5B	01	B3	5A
00004448	00	B3	5A	01	B3	5A	01	B3	5A	01	B3	5A	01	B3	5A	00
00004464	B3	5A	01	B3	5A	01	B2	5B	00	B2	5B	01	B2	5B	00	B2
00004480	5B	00	B2	5B	00	B3	5C	01	B3	5C	00	B3	5C	01	B2	5C
00004496	00	B2	5C	00	B3	5D	01	B3	5D	01	B3	5D	00	B3	5D	01
00004512	B4	5E	00	B4	5E	00	B5	5F	00	B5	5F	01	B5	5F	01	B5
00004528	5F	01	B5	5F	00	B5	5F	00	B5	5F	01	B5	5F	01	B3	5E
00004544	00	B3	5E	01	B2	5D	00	B2	5D	01	B2	5D	01	B2	5D	00
00004560	B1	5C	00	B1	5C	01	B4	5E	00	B4	5E	01	B3	5D	00	B3
00004576	5C	01	B2	5B	01	B1	59	01	B0	58	01	B0	58	01	B1	59
00004592	00	B1	59	01	B1	59	00	B1	59	01	B1	59	00	B1	59	00
00004608	B1	59	01	B1	59	01	B1	59	00	B1	59	01	B1	59	01	B1
00004624	59	01	B1	59	00	B1	59	01	B1	59	00	B1	59	00	B1	59
00004640	00	B1	59	01	B1	59	00	B1	59	00	B1	59	00	B1	59	01
00004656	B1	59	00	B1	59	01	B1	5C	00	B1	5B	01	B0	5C	01	B1
00004672	5C	00	AF	5D	00	B0	5D	01	AF	5E	01	AF	5E	01	AA	5D
00004688	00	AD	5F	00	B0	5F	01	B2	61	01	B4	60	00	B7	5F	01
00004704	B7	5E	00	B6	5E	00	B3	5B	00	B3	5B	01	B3	5B	01	B2
00004720	5A	00	B2	5A	01	B1	59	01	B1	59	01	B1	59	00	B2	5A
00004736	00	B2	5A	00	B2	5A	01	B2	5A	01	B2	5A	00	B2	5A	00
00004752	B2	5A	00	B2	5A	00	B2	5A	00	B2	5A	01	B2	5A	01	B2
00004768	5A	00	B2	5A	00	B2	5A	01	B2	5A	01	B2	5A	01	B2	5A
00004784	00	B2	5A	01	B2	5A	00	B2	5A	01	B2	5A	00	B2	5A	00
00004800	B2	5A	01	B2	5A	00	B2	5A	00	B2	5A	00	B2	5A	01	B2
00004816	5A	01	B2	5A	00	B2	5A	01	B2	5A	00	B2	5A	00	B1	59
00004832	00	B1	59	01	B1	59	01	B1	59	01	B1	59	00	B1	59	00
00004848	B1	59	00	B1	59	00	B1	59	00	B1	59	01	B1	59	00	B1
00004864	59	00	B1	59	01	B1	59	00	B1	59	00	B1	59	00	B2	5A
00004880	00	B2	5A	01	B2	5A	01	B2	5A	01	B2	5A	01	B2	5A	00
00004896	B2	5A	00	B2	5A	01	B2	5A	00	B2	5A	01	B2	5A	01	B2
00004912	5A	01	B2	5A	01	B2	5A	01	B2	5A	00	B2	5A	01	B2	5A
00004928	31	B2	5A	31	B2	5A	31	B2	5A	31	B2	5A	31	B2	5A	31
00004944	B2	5A	31	B2	5A	31	B2	5A	31	B2	5A	31	B2	5A	31	B2
00004960	5A	31	B2	5A	31	B2	5A	31	B2	5A	31	B2	5A	31	B1	5B

就是这些乱七八糟的
把这些抠出来，仔细观察一下，看！
（・(ェ)・）：看什么？
有没有发现这是RGB（Σ(°△°|||)﴾ ，这个写writeup一定是个弱智算了我们不看了）
在R的位置上呈现的是二进制

```
00 B6 61 01 B6 61 00 B5 60 00 B5 60 01 B5 60
B5 60 00 B4 5F 01 B4 5F 00 B4 5F 01 B4 5F 00
5E 01 B3 5E 00 B3 5E 00 B3 5E 01 B3 5E 01 B1
00 B0 5E 01 B2 5D 00 B2 5D 00 B2 5C 00 B3 5B
B3 5B 01 B3 5B 01 B3 5A 00 B3 5A 01 B3 5A 01
5A 01 B3 5A 01 B3 5A 00 B3 5A 01 B3 5A 01 B2
00 B2 5B 01 B2 5B 00 B2 5B 00 B2 5B 00 B3 5C
B3 5C 00 B3 5C 01 B2 5C 00 B2 5C 00 B3 5D 01
5D 01 B3 5D 00 B3 5D 01 B4 5E 00 B4 5E 00 B5
00 B5 5F 01 B5 5F 01 B5 5F 01 B5 5F 00 B5 5F
B5 5F 01 B5 5F 01 B3 5E 00 B3 5E 01 B2 5D 00
5D 01 B2 5D 01 B2 5D 00 B1 5C 00 B1 5C 01 B4
00 B4 5E 01 B3 5D 00 B3 5C 01 B2 5B 01 B1 59
B0 58 01 B0 58 01 B1 59 00 B1 59 01 B1 59 00
59 01 B1 59 00 B1 59 00 B1 59 01 B1 59 01 B1
00 B1 59 01 B1 59 01 B1 59 01 B1 59 00 B1 59
B1 59 00 B1 59 00 B1 59 00 B1 59 01 B1 59 00
59 00 B1 59 00 B1 59 01 B1 59 00 B1 59 01 B1
00 B1 5B 01 B0 5C 01 B1 5C 00 AF 5D 00 B0 5D
AF 5E 01 AF 5E 01 AA 5D 00 AD 5F 00 B0 5F 01
61 01 B4 60 00 B7 5F 01 B7 5E 00 B6 5E 00 B3
00 B3 5B 01 B3 5B 01 B2 5A 00 B2 5A 01 B1 59
B1 59 01 B1 59 00 B2 5A 00 B2 5A 00 B2 5A 01
5A 01 B2 5A 00 B2 5A 00 B2 5A 00 B2 5A 00 B2
00 B2 5A 01 B2 5A 01 B2 5A 00 B2 5A 00 B2 5A
B2 5A 01 B2 5A 01 B2 5A 00 B2 5A 01 B2 5A 00
5A 01 B2 5A 00 B2 5A 00 B2 5A 01 B2 5A 00 B2
00 B2 5A 00 B2 5A 01 B2 5A 01 B2 5A 00 B2 5A
B2 5A 00 B2 5A 00 B1 59 00 B1 59 01 B1 59 01
59 01 B1 59 00 B1 59 00 B1 59 00 B1 59 00 B1
00 B1 59 01 B1 59 00 B1 59 00 B1 59 01 B1 59
B1 59 00 B1 59 00 B2 5A 00 B2 5A 01 B2 5A 01
5A 01 B2 5A 01 B2 5A 00 B2 5A 00 B2 5A 01 B2
00 B2 5A 01 B2 5A 01 B2 5A 01 B2 5A 01 B2 5A
B2 5A 00 B2 5A 01 B2 5A
```

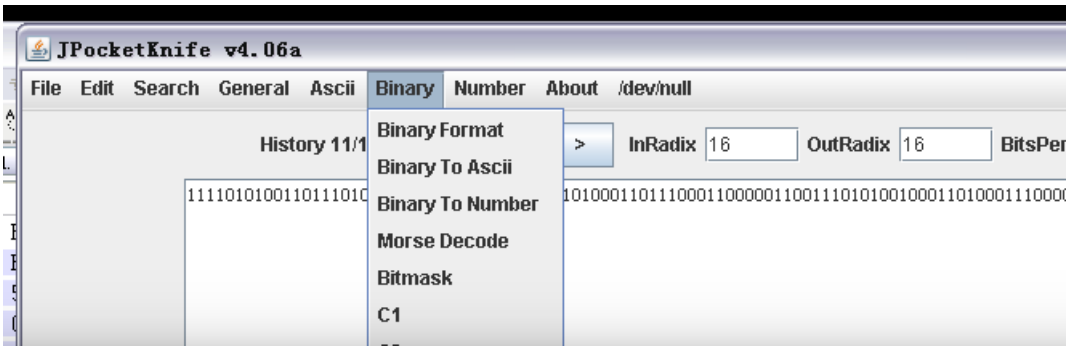
所以接下来就是把R位置上的数字挑出来，虽然没多少但是这种重复的工作就交给代码吧

Start here | *func.cpp

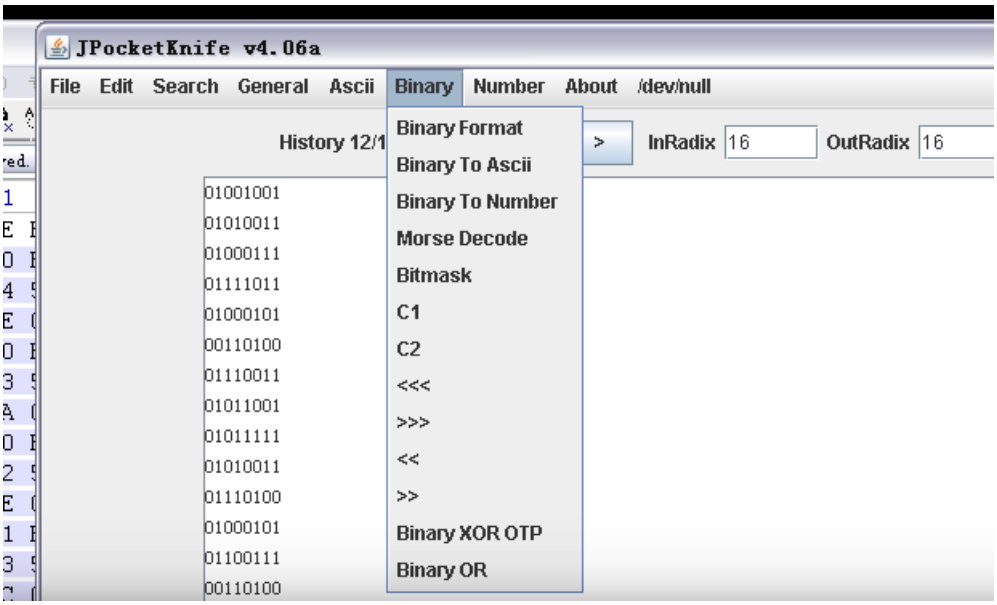
```cpp
#include <iostream>
#include <cstdio>
#include <cstring>
#include <algorithm>
using namespace std;

int main()
{
    char temp[100];
    int len;
    freopen("asc1","r",stdin);
    freopen("binary","w",stdout);
    while(gets(temp)!=NULL)
    {
        len = strlen(temp);
        for(int i=0;i<len-1;i++)
        {
            if(temp[i] == '0' && temp[i+1] == '0')
                printf("0");
            else if(temp[i] == '0' && temp[i+1] == '1')
                printf("1");
            else
                printf("");
        }
    }
    return 0;
}
```

然后处理完就是这样的

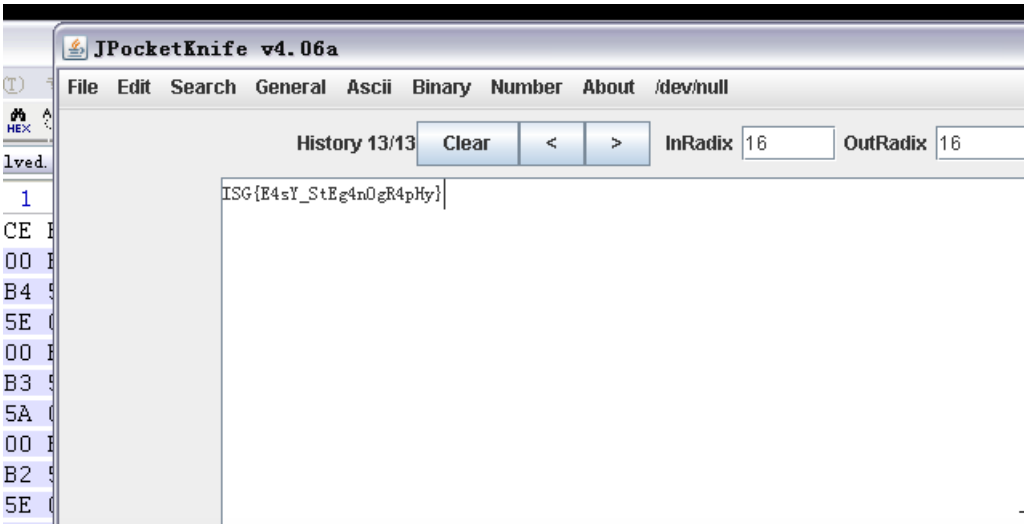
其实最开始我是保留所有的，也就是00 直接保留00，01直接保留01，但是后来解码发现有错误，然后就把00改为0，01改为1
从小书包掏出stegsolve的好朋友JPocketKnife
两部曲：先binary format



然后to ascii



然后就没了，真的没了还看啥？



这一题参考了两个大佬的解题思路，总结了一下，写出的博文

如果有什么错误的地方请你们指正，谢谢

您可以考虑给博主来个小小的打赏以资鼓励，您的肯定将是我最大的动力。


作者： 落花四月
出处： https://www.cnblogs.com/lxz-1263030049/
关于作者：潜心于网络安全学习。如有问题或建议，请多多赐教！
版权声明：本文版权归作者和博客园共有，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文连接.
特此声明：所有评论和私信都会在第一时间回复。也欢迎园子的大大们指正错误，共同进步。或者直接私信我
声援博主：如果您觉得文章对您有帮助，可以点击文章右下角【推荐】一下。您的鼓励是作者坚持原创和持续写作的最大动力！