




实验吧安全杂项WP(三)

原创

[Neil-Yale](#)  于 2017-03-20 21:30:46 发布  5244  收藏 2

文章标签: [c语言](#) [安全](#) [百度](#) [压缩](#) [CTF](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/yalecaltech/article/details/64158016>

版权

1.有趣的文件 (<http://www.shiyanbar.com/ctf/1861>)

最前面的8位是地址，不用管，后面的应该是文件头，百度afbc 1c27



The screenshot shows a Baidu search interface. The search bar contains 'afbc 1c27'. Below the search bar, there are navigation tabs for '网页', '新闻', '贴吧', '知道', '音乐', '图片', '视频', '地图', '文库', and '更多'. The search results indicate approximately 28,600,000 related results. A filter option '您可以仅查看: 英文结果' is visible. The top result is a link to '7-Zip / Discussion / Help:Error: Can not open file as archive'. Below the link, there is a snippet of text: '查看此网页的中文翻译, 请点击 翻译此页' followed by a hex dump: '00000000 37 7A BC AF 27 1C 00 02 A2 57 93 64 A4 5C 3E D6 00000010 1F 00 00 00 2 A 00 00 00 00 00 00 1E EB 0F AC ...'. The snippet also includes a rating: 'sourceforge.net/p/seve... - 百度快照 - 86%好评'.

看样子应该是.7z的压缩文件，不过给出的ascii里面没有37 7a，这就是缺少的，需要我们补上，补上后发现什么文件也不是，问题出在哪里呢？
百度后发现每两位应该交换一下



The screenshot shows a Baidu search interface. The search bar contains '7z的文件头afbc 1c27'. Below the search bar, there are navigation tabs for '网页', '新闻', '贴吧', '知道', '音乐', '图片', '视频', '地图', '文库', and '更多'. The search results indicate approximately 28 related results. A filter option '您可以仅查看: 英文结果' is visible. The top result is a link to '相同的7z文件为什么不能解压? 百度知道'. Below the link, there is a snippet of text: '1个回答 - 提问时间: 2012年06月04日' followed by: '7z文件正确的头部特征码:37 7A BC AF 27 1C 00 03这两个文件大小相同,我用了CRC32、CRC16、MD5、SHA256校验方式(我都测了好几十次了),结果这两个文件的...'. The snippet also includes a rating: '更多关于7z的文件头afbc 1c27的问题>>' and 'zhidao.baidu.com/link? - 百度快照 - 评价'.

这个任务太繁重了，本来还想这放在winhex里面手工的，这里直接python吧

```
def revStr(s):  
    news=""  
    for i in xrange(0,len(s),4):  
        news+=s[i+2:i+4]  
        news+=s[i:i+2]  
    return news
```

```

def foo():
f=open('funfile')
s="377a"
for line in f:
s+=revStr(line.strip())[8:].replace(' ',';')
fsave=open('fun.7z','wb')
fsave.write(s.decode('hex'))
fsave.close()
pass
if name == 'main':
foo()
print 'finished'

```

自动生成fun.7z压缩文件，解压后是一张阿狸的图片，拖进winhex看看，发现疑似flag的base64加密过的

00000050	17 18 16 14 18 12 14 15	14 FF DB 00 43 01 03 04	yÛ C
00000060	04 05 04 05 09 05 05 09	14 0D 0B 0D 2F 2E 2A 2F	/*
00000070	65 5E 40 20 65 76 61 6C	28 65 63 68 6F 28 20 51	e^@ eval(echo(Q
00000080	31 52 47 7B 52 6C 56 4F	54 6C 6C 66 55 44 46 44	1RG{R1VOT11fUDFD
00000090	56 46 56 53 4D 77 3D 3D	7D 14 14 14 14 14 FF C0	VFVSMw==} yÀ
000000A0	00 11 08 01 F0 02 6B 03	01 11 00 02 11 01 03 11	ø k
000000B0	01 FF C4 00 1E 00 00 02	01 05 01 01 01 00 00 00	yÀ

复制后base64解码就行了

2.Paint&Scan(<http://www.shiyanbar.com/ctf/1860>)

只给出了坐标，肯定是要作图的咯

先把数据保存下来，然后跑python

(很多大神用Matlab,但是体积太大；有用matplotlib库的，但是安装麻烦)



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
PythonPractice 1.py
This file is indented with 2 spaces instead of 4
1 import sys
2 from PIL import Image
3 fin=open("1.txt","r")
4 s=s.stdin=fin
5 size=(500,500)
6 img=Image.new("RGB",size,(255,255,255))
7 data=img.getdata()
8 xy=(0,0)
9 while True:
10     try:
11         xy=input()
12         data.putpixel(xy,(0,0,0))
13     except:
14         break
Run 1
C:\Python27\python.exe C:/Users/hasee/Desktop/Pytho
Process finished with exit code 0
```

a.jpg - 照片
查看所有照片 共享 缩放 绘制



<http://blog.csdn.net/yalecaltech>

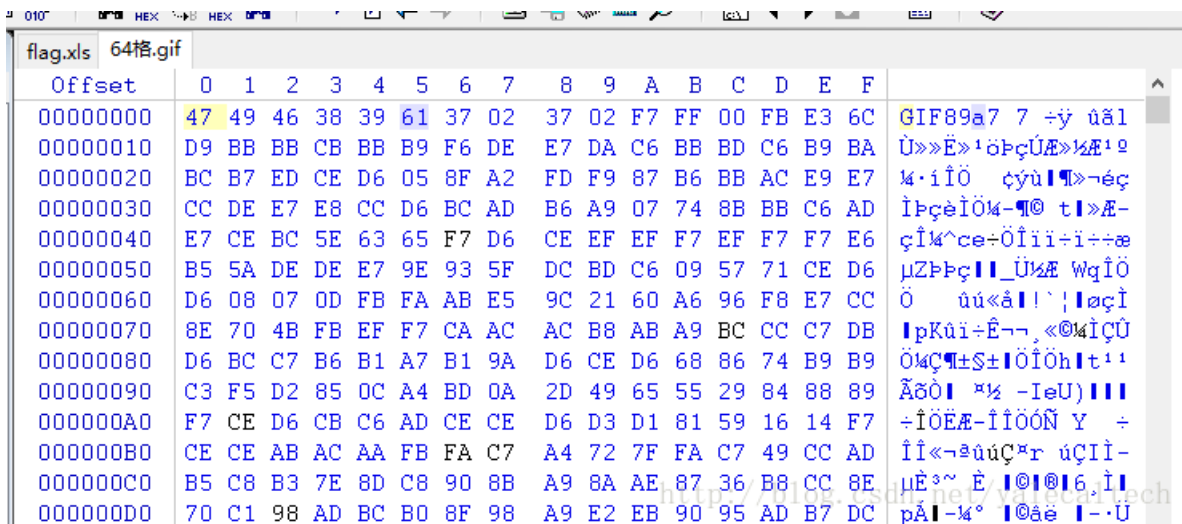
扫描就得到结果了

3.64格 (<http://www.shiyanbar.com/ctf/1857>)

下载来的文件解压后是gif，打不开，直接仍winhex

发现少文件头

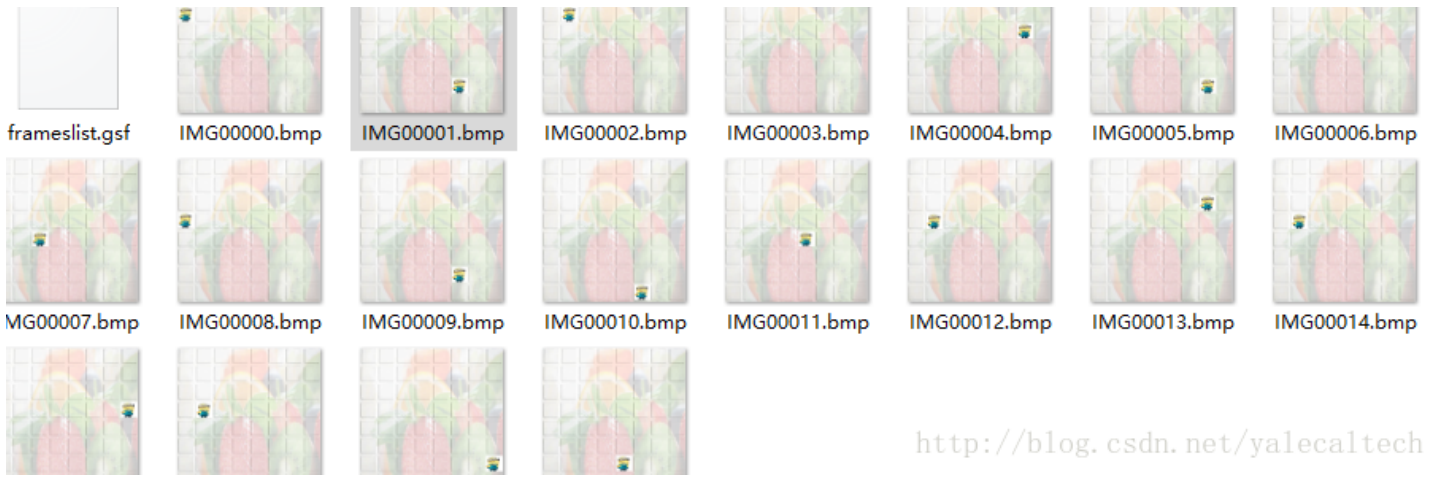
补上



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	47	49	46	38	39	61	37	02	37	02	F7	FF	00	FB	E3	6C	GIF89a7 7 +y uäl
00000010	D9	BB	BB	CB	BB	B9	F6	DE	E7	DA	C6	BB	BD	C6	B9	BA	Ù»»È»'ôpçÚÆ»½Æ¹²
00000020	BC	B7	ED	CE	D6	05	8F	A2	FD	F9	87	B6	BB	AC	E9	E7	¼·iîÖ çyù!¶»-éc
00000030	CC	DE	E7	E8	CC	D6	BC	AD	B6	A9	07	74	8B	BB	C6	AD	İpçèİÖ¼-¶@ t »Æ-
00000040	E7	CE	BC	5E	63	65	F7	D6	CE	EF	EF	F7	EF	F7	F7	E6	çİ¼^ce÷Öiîi+i÷÷æ
00000050	B5	5A	DE	DE	E7	9E	93	5F	DC	BD	C6	09	57	71	CE	D6	µZpç _Ù½Æ WqİÖ
00000060	D6	08	07	0D	FB	FA	AB	E5	9C	21	60	A6	96	F8	E7	CC	Ö úú«â! ` lœçİ
00000070	8E	70	4B	FB	EF	F7	CA	AC	AC	B8	AB	A9	BC	CC	C7	DB	İpKûi÷È-¶-«@¼içÜ
00000080	D6	BC	C7	B6	B1	A7	B1	9A	D6	CE	D6	68	86	74	B9	B9	Ö¼ç¶±\$±İÖİöhit¹¹
00000090	C3	F5	D2	85	0C	A4	BD	0A	2D	49	65	55	29	84	88	89	ÃðÖ! ½ -İeU)!!!
000000A0	F7	CE	D6	CB	C6	AD	CE	CE	D6	D3	D1	81	59	16	14	F7	÷iÖÈÆ-İiÖÓN Y ÷
000000B0	CE	CE	AB	AC	AA	FB	FA	C7	A4	72	7F	FA	C7	49	CC	AD	İİ«-ªúç*ªr úçİi-
000000C0	B5	C8	B3	7E	8D	C8	90	8B	A9	8A	AE	87	36	B8	CC	8E	µÈ³~ È @ @ 6.İ
000000D0	70	C1	98	AD	BC	B0	8F	98	A9	E2	EB	90	95	AD	B7	DC	pÄ -¼° @âè --U

保存后打开，发现是动图，使用gifsplitter分离得到十九张图片





<http://blog.csdn.net/yalecaltech>

题目提示64格，图片刚好也是64格

64联想到64进制？base64？

百度64进制

右边这幅图好像有点感觉

六十四进制

[编辑](#)

[+](#) [★ 收藏](#) [👍 49](#) [🔗 0](#)

Base64是一种基于64个可打印字符来表示二进制数据的表示方法。由于2的6次方等于64，所以每6个位元为一个单元，对应某个可打印字符。三个字节有24个位元，对应于4个Base64单元，即3个字节需要用4个可打印字符来表示。它可用来作为电子邮件的传输编码。在Base64中的可打印字符包括字母A-Z、a-z、数字0-9，这样共有62个字符，此外两个可打印符号在不同的系统中而不同。一些如uuencode的其他编码方法，和之后binhex的版本使用不同的64字符集来代表6个二进制数字，但是它们不叫Base64。

☰坤	☱剥	☰比	☱观	☱豫	☱晋
0	1	2	3	4	5
☱谦	☱艮	☱蹇	☱渐	☱恒	☱旅
8	9	10	11	12	13
☱师	☱蒙	☱坎	☱涣	☱解	☱睽
16	17	18	19	20	21
☱升	☱蛊	☱井	☱巽	☱恒	☱鼎
24	25	26	27	28	29
☱复	☱颐	☱屯	☱益	☱震	☱贲
32	33	34	35	36	37
☱夬	☱贲	☱蹇	☱兑	☱丰	☱离
40	41	42	43	44	45

对应这维尼所在的格子求得18个数字

16, 53, 17,

再将数字对应百度百科下的表解得字母



六十四进制



2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8

12	m
13	N
14	O
15	P

28	c
29	d
30	e
31	f

44	s
45	t
46	u
47	v

60	u
61	9
62	+
63	/

<http://blog.csdn.net/yalecaltech>

得到Q1RGe2FiY19kZWZfZ30

base64解码之

得到CTF{abc_def_g}