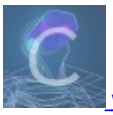
Bugku CTF 杂项（21-29） Writeup

原创

KRDecad3 于 2018-06-23 16:57:32 发布 4004 收藏 11

分类专栏： writeup 文章标签： Bugku CTF writeup

本文链接：https://blog.csdn.net/KRDecad3/article/details/80784989



Bugku CTF 杂项（21-29） Writeup

0x21图穷匕见

下载得到一个图片，用winhex打开发现jpg文件尾FF D8后面有大量16进制数据，复制后面的数据，用notepad++中插件convert，hex->ascii转换后是一个个坐标点。
将括号和逗号去掉保存为txt文件。

```
7 7
7 8
7 9
7 10
7 11
7 12
7 13
7 14
7 15
7 16
7 17
7 18
7 19
7 20
7 21
7 22
7 23
```


再利用gnuplot画图（windows），输入：“plot “文件名””，回车得到一张二维码（注意，文件路径的反斜杠要转义）。
不知为啥，自己画出来的扫描不出来。

0x22convert

convert转换的意思，打开是一串二进制，将它转换成十六进制，再把十六进制字符写到HxD中，发现文件头是52 61 72，rar文件头。就把它保存成rar文件，解压得到一张图片，查看属性，里面有一串base64编码，解码得到flag。

```
52 61 72 21 1A 07 00 CF 90 73 00 00 0D 00 00 00  Rar!...Ï.s......
00 00 00 00 F7 C0 74 20 90 2C 00 0D 09 00 00 59  ....÷Àt .,.....Y
22 00 00 02 3E 63 70 19 0A 59 B3 4A 1D 33 07 00  "...>cp..Y³J.3..
```


附上一个脚本：

```python
import binascii

__author__ = 'feifei'
# !/usr/bin/env python
# -*- coding: utf-8 -*-


base = [str(x) for x in range(10)] + [chr(x) for x in range(ord('A'), ord('A') + 6)]


# bin2dec
def bin2dec(string_num):
    return str(int(string_num, 2))


# hex2dec
def hex2dec(string_num):
    return str(int(string_num.upper(), 16))


# dec2bin
def dec2bin(string_num):
    num = int(string_num)
    mid = []
    while True:
        if num == 0: break
        num, rem = divmod(num, 2)
        mid.append(base[rem])

    return ''.join([str(x) for x in mid[::-1]])


# dec2hex
def dec2hex(string_num):
    num = int(string_num)
    mid = []
    while True:
        if num == 0:
            break
        num, rem = divmod(num, 16)
        mid.append(base[rem])

    return ''.join([str(x) for x in mid[::-1]])


# hex2tobin
def hex2bin(string_num):
    return dec2bin(hex2dec(string_num.upper()))
```

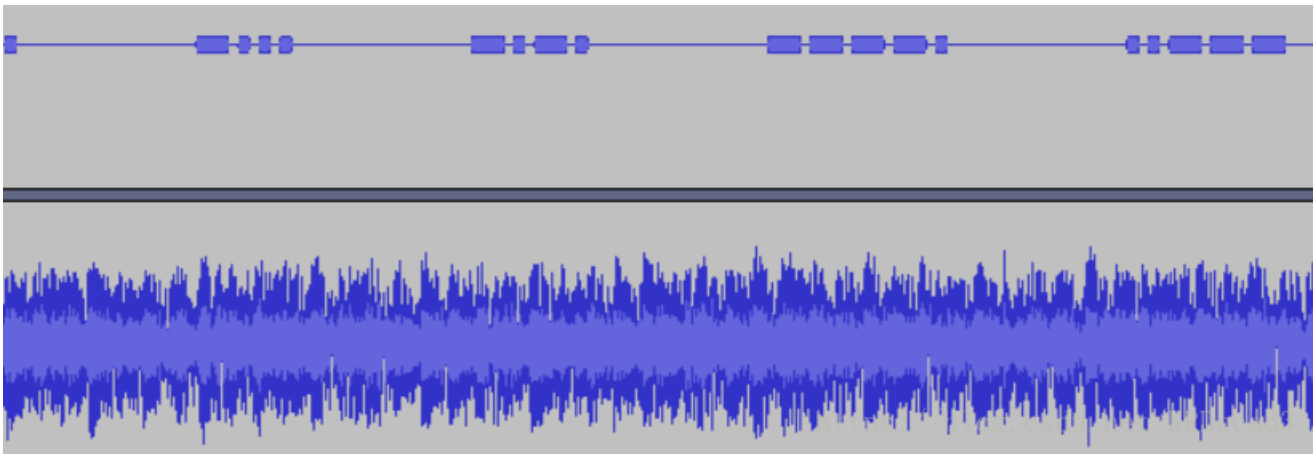
```
# bin2hex
def bin2hex(string_num):
    return dec2hex(bin2dec(string_num))


if __name__ == '__main__':
    file1 = open('convert.txt')
    s = file1.read()
    hexx = bin2hex(s)
    print hexx
    file2 = open('4.rar', 'wb')
    file2.write(binascii.a2b_hex(hexx))
```

来自https://blog.csdn.net/yaofeiNO1/article/details/78459569#t3

0x23听首音乐

下载得到一个音频，用Audacity音频分析软件打开，



猜测是摩尔斯电码，长的用“-”表示，短的用“.”表示，中间用空格隔开。

```
.....  -...  -.-.  ----.  ..---  .....  -....  ....-  ----.  -.-.  -...  -----  .----  ---..
---..  ..-.  .....  ..---  .  -....  .----  --...  -..  --...  -----  ----.  ..---  ----.  .----
----.  .----  -.-.
```


解密得到答案。

0x24好多数值

打开发现像是坐标一样的东西，并且看到数值255，猜测与RGB有关。

```
255, 255, 255
255, 255, 255
255, 255, 255
255, 255, 255
255, 255, 255
255, 255, 255
255, 255, 255
255, 255, 255
255, 255, 255
255, 255, 255
255, 255, 255
255, 255, 255
```


需要用到python的PIL库，知识储备不足。。。
RGB值转化图片（python PIL）https://www.cnblogs.com/webFuckeeeer/p/4536776.html

---

0x25很普通的数独

下载下来有25张数独图片，网上说按5*5排列是个二维码，但是第1张是二维码右上角，第5张是二维码左下角，第21张是二维码左上角。
位置调好后，将带数字的用0表示，空白处用1表示，再用脚本生成二维码。

0x26好多压缩包

解压后发现有68个压缩包，而且每个压缩包里的txt文件都有密码，这里用到一个知识“CRC32碰撞”https://www.anquanke.com/post/id/86211。
写一个脚本进行暴破（python3）

```python
import zipfile
import string
import binascii

def CrackCrc(crc):
    for i in dic: #迭代的不是值而是键（key）
        for j in dic:
            for k in dic:
                for h in dic:
                    s = i + j + k + h
                    if crc == (binascii.crc32(s.encode())):
                        f.write(s)
                        return

def CrackZip():
        for i in range(0,68):
            file = 'out'+str(i)+'.zip'
            crc = zipfile.ZipFile(file,'r').getinfo('data.txt').CRC
            CrackCrc(crc)

dic = string.ascii_letters + string.digits + '+/='

f = open('out.txt','w')
CrackZip()
print("CRC32碰撞完成")
f.close
```

运行后得到一个out.txt文件，里面是一串base64，解码得到16进制，

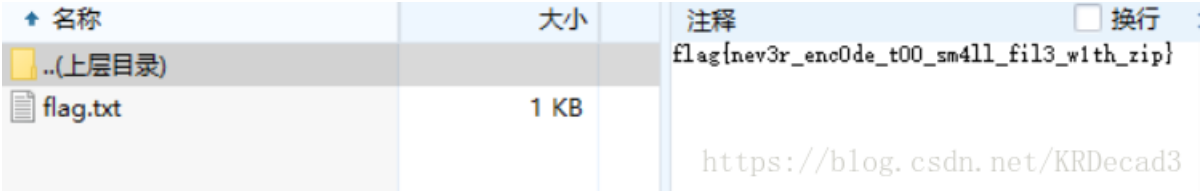
```
z5BzAAANAAAAAAAAAKo+egCAIwBJAAAAVAAAAAKGNKv
+a2MdSR0zAwABAAAAQ01UCRUUy91BT5UkSNPoj5hFEVFBRvefHSBCfG0ruGnKnygsMy
EZ24cXtZ01y3k1K1YJ0vpK9HwqUzb6u9z8igEr3dCCQLQAdAAAAHQAAAAJi0efVT2Md
hZy50eHQAsDRpZmZpeCB0aGUgZm1sZSBhbmQgZ2V0IHRoZSBmbGFnxD17AEAHAA==
```

```
cf 90 73 00 00 0d 00 00
00 00 00 00 00 aa 3e 7a
00 80 23 00 49 00 00 00
54 00 00 00 02 86 34 ab
fe 6b 63 1d 49 1d 33 03
00 01 00 00 00 43 4d 54
09 15 14 cb dd 41 4f 95
24 48 d3 e8 8f 98 45 11
51 41 46 f7 9f 1d 20 42
7c 6d 2b b8 69 ca 9f 28
2c 33 28 fc 48 16 99 1f
1b 18 1d 8f 38 2c 46 76
```

复制到HxD中，发现底部有rar的文件尾C4 3D 7B 00 40 07 00，还存在一个名为CMT即comment的文件，

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
52 61 72 21 1A 07 00 CF 90 73 00 00 0D 00 00 00  Rar!...Ï.s......
00 00 00 00 AA 3E 7A 00 80 23 00 49 00 00 00 54  ....ª>z.€#.I...T
00 00 00 02 86 34 AB FE 6B 63 1D 49 1D 33 03 00  ....†4«þkc.I.3..
01 00 00 00 43 4D 54 09 15 14 CB DD 41 4F 95 24  ....CMT...ËÝAO•$
48 D3 E8 8F 98 45 11 51 41 46 F7 9F 1D 20 42 7C  HÓè.˜E.QAF÷Ÿ. B|
6D 2B B8 69 CA 9F 28 2C 33 28 FC 48 16 99 1F 1B  m+¸iÊŸ(,3(üH.™..
18 1D 8F 38 2C 46 76 E1 C5 ED 67 4D 72 DE 4D 4A  ...8,FváÅígMrÞMJ
D5 82 74 BE 92 BD 1F 0A 94 CD BE AE F7 3F 22 80  Õ‚t¾'½..”Í¾®÷?"€
4A F7 74 20 90 2D 00 1D 00 00 00 1D 00 00 00 02  J÷t .-..........
62 D1 E7 D5 4F 63 1D 49 1D 30 08 00 20 00 00 00  bÑçÕOc.I.0.. ...
66 6C 61 67 2E 74 78 74 00 B0 34 69 66 66 69 78  flag.txt.°4iffix
20 74 68 65 20 66 69 6C 65 20 61 6E 64 20 67 65   the file and ge
74 20 74 68 65 20 66 6C 61 67 C4 3D 7B 00 40 07  t the flagÄ={.@.
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

并且提示“fix the file and get the flag”，说明要修复文件，把rar文件头52 61 72 21 1A 07 00补上，保存成rar文件，用解压软件打开，在注释里找到flag。



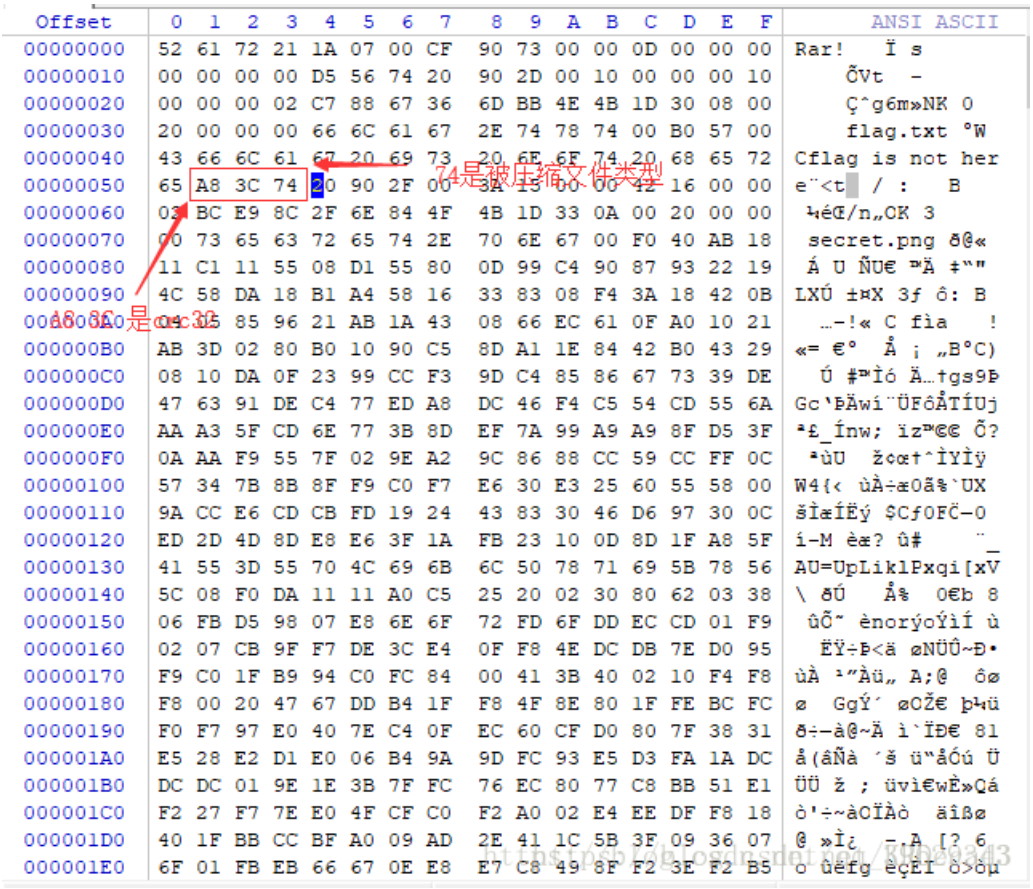
0x27一个普通的压缩包(xp0intCTF)

下载解压得到一个flag.txt打开写着flag不在里面。
用HxD打开压缩包，发现文件头PK，修改后缀为zip解压，

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
```

```
50 4B 03 04 0A 00 00 08 00 00 E7 AB 51 4B 00 00  PK........ç«QK..
00 00 00 00 00 00 00 00 00 00 19 00 00 00 E4 B8  ..............ä¸
80 E4 B8 AA E6 99 AE E9 80 9A E7 9A 84 E5 8E 8B  €ä¸ªæ™®é€šçš„åŽ‹
E7 BC A9 E5 8C 85 2F 50 4B 03 04 0A 00 00 08 00  ç¼©åŒ…/PK.......
00 AD 85 4F 4B E4 8D 76 94 C1 15 00 00 C1 15 00  ...…OKä.v”Á...Á..
00 21 00 00 00 E4 B8 80 E4 B8 AA E6 99 AE E9 80  .!...ä¸€ä¸ªæ™®é€
9A E7 9A 84 E5 8E 8B E7 BC A9 E5 8C 85 2F 66 6C  šçš„åŽ‹ç¼©åŒ…/fl
61 67 2E 72 61 72 52 61 72 21 1A 07 00 CF 90 73  ag.rarRar!...Ï.s
00 00 0D 00 00 00 00 00 00 00 D5 56 74 20 90 2D  ..........ÕVt .-
```

里面有一个flag.rar，我并没有遇到他们说的解压打不开的情况。
用HxD修改一下flag.rar



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
00000000	52	61	72	21	1A	07	00	CF	90	73	00	00	0D	00	00	00	Rar! Ï s
00000010	00	00	00	00	D5	56	74	20	90	2D	00	10	00	00	00	10	ÕVt -
00000020	00	00	00	02	C7	88	67	36	6D	BB	4E	4B	1D	30	08	00	Ç^g6m»NK 0
00000030	20	00	00	00	66	6C	61	67	2E	74	78	74	00	B0	57	00	flag.txt °W
00000040	43	66	6C	61	67	20	69	73	20	6E	6F	74	20	68	65	72	Cflag is not her
00000050	65	A8	3C	74	20	90	2F	00	3A	15	00	00	42	16	00	00	e¨<t / : B
00000060	03	BC	E9	8C	2F	6E	84	4F	4B	1D	33	0A	00	20	00	00	¼éŒ/n„OK 3
00000070	00	73	65	63	72	65	74	2E	70	6E	67	00	F0	40	AB	18	secret.png ð@«
00000080	11	C1	11	55	08	D1	55	80	0D	99	C4	90	87	93	22	19	Á U ÑU€ ™Ä ‡“"
00000090	4C	58	DA	18	B1	A4	58	16	33	83	08	F4	3A	18	42	0B	LXÚ ±¤X 3ƒ ô: B
000000A0	A8	3C	04	85	96	21	AB	1A	43	08	66	EC	61	0F	A0	10	21	¨<…–!« C fìa !
000000B0	AB	3D	02	80	B0	10	90	C5	8D	A1	1E	84	42	B0	43	29	«= €° Å ¡ „B°C)
000000C0	08	10	DA	0F	23	99	CC	F3	9D	C4	85	86	67	73	39	DE	Ú #™Ìó Ä…†gs9Þ
000000D0	47	63	91	DE	C4	77	ED	A8	DC	46	F4	C5	54	CD	55	6A	Gc'ÞÄwí¨ÜFôÅTÍUj
000000E0	AA	A3	5F	CD	6E	77	3B	8D	EF	7A	99	A9	A9	8F	D5	3F	ª£_Ínw; ïz™©© Õ?
000000F0	0A	AA	F9	55	7F	02	9E	A2	9C	86	88	CC	59	CC	FF	0C	ªùU žc¢œ†ˆÌYÌÿ
00000100	57	34	7B	8B	8F	F9	C0	F7	E6	30	E3	25	60	55	58	00	W4{‹ ùÀ÷æ0ã%`UX
00000110	9A	CC	E6	CD	CB	FD	19	24	43	83	30	46	D6	97	30	0C	šÌæÍËý $Cƒ0FÖ—0
00000120	ED	2D	4D	8D	E8	E6	3F	1A	FB	23	10	0D	8D	1F	A8	5F	í-M èæ? û# ¨_
00000130	41	55	3D	55	70	4C	69	6B	6C	50	78	71	69	5B	78	56	AU=UpLiklPxqi[xV
00000140	5C	08	F0	DA	11	11	A0	C5	25	20	02	30	80	62	03	38	\ ðÚ Å% 0€b 8
00000150	06	FB	D5	98	07	E8	6E	6F	72	FD	6F	DD	EC	CD	01	F9	ûÕ˜ ènoryoÝìÍ ù
00000160	02	07	CB	9F	F7	DE	3C	E4	0F	F8	4E	DC	DB	7E	D0	95	ËŸ÷Þ<ä øNÜÛ~Ð•
00000170	F9	C0	1F	B9	94	C0	FC	84	00	41	3B	40	02	10	F4	F8	ùÀ ¹”Àü„ A;@ ôø
00000180	F8	00	20	47	67	DD	B4	1F	F8	4F	8E	80	1F	FE	BC	FC	ø GgÝ´ øOŽ€ þ¼ü
00000190	F0	F7	97	E0	40	7E	C4	0F	EC	60	CF	D0	80	7F	38	31	ð÷—à@~Ä ì`ÏÐ€ 81
000001A0	E5	28	E2	D1	E0	06	B4	9A	9D	FC	93	E5	D3	FA	1A	DC	å(âÑà ´š ü“åÓú Ü
000001B0	DC	DC	01	9E	1E	3B	7F	FC	76	EC	80	77	C8	BB	51	E1	ÜÜ ž ; üvì€wÈ»Qá
000001C0	F2	27	F7	7E	E0	4F	CF	C0	F2	A0	02	E4	EE	DF	F8	18	ò'÷~àOÏÀò äîßø
000001D0	40	1F	BB	CC	BF	A0	09	AD	2E	41	1C	5B	3F	09	36	07	@ »Ì¿ .A [? 6
000001E0	6F	01	FB	EB	66	67	0E	E8	E7	C8	49	8F	F2	3E	F2	B5	o ûëfg èçÈI ò>òµ

解压有一个png文件，用HxD查看发现是gif图片，更改后缀，用stegsolve工具打开，只能找到二维码的下半部分。

用gifsplitter工具发现这个gif是两帧，并将gif分离，使用stegsolve工具打开这两张图片，然后用PS合并补全，扫一扫。
ps:之前以为把上下两半拼起来再把左下角复制一块补到左上角，这样做一直没扫出来，后来才发现右上角也得补一块/xk。

0x28妹子的陌陌

保存图片，在kali里用binwalk分析，看到有rar文件，foremost分离，解压发现有密码，没想到的是，密码居然在图片上写着。
解压后得到一个momo.txt文件，

```
嘟嘟嘟嘟
士兵：报告首长！已截获纳粹的加密电报！
首长：拿来看看

电报内容：
..../-/-/.--./---.../-.-.-/-.-.-./.-/-.-.-./---/-.././.-.-.-/-.-./..../.-/...
---/.-.-.-/-.-.-/---/--/-..-.

首长：我操你在逗我吗？你确定是他们纳粹发的吗？
士兵：难道我弄错了？哦。。。等等是这一条

内容：http://c.bugku.com/U2FsdGVkX18t18Yi7FaGiv6jK1SBxKD30eYb52onYe0=
      AES Key：@#@#￥%……￥￥%%……&￥

士兵：二维码真的扫不出来吗？？肯定可以扫出来
```


解密莫尔斯电码，得到一个在线解密的网址，再解下面的AES，最后打开页面得到一个二维码，使用stegsolve进行反色处理，扫一扫。

0x29就五层你能解开吗

提示：第一层：CRC32 碰撞

第二层：维吉尼亚密码

第三层：sha1 碰撞

第四层：md5 相同文件不同

第五层：RSA

下载压缩包，用解压软件打开；

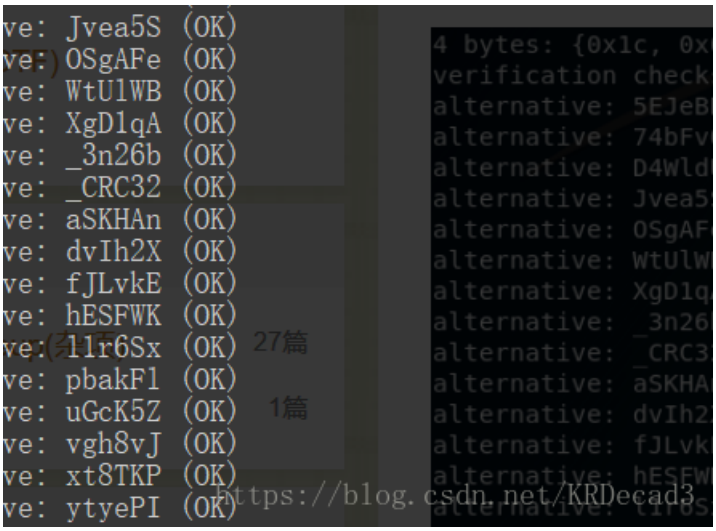
名称	大小	压缩后大小	类型	安全	修改时间	CRC32	压缩算法
..(上层目录)							
CRC32 Collision.7z *	246.76 KB	246.79 KB	好压 7Z 压缩文件		2016-10-08 01:59:	06B072C5	LZMA2:18 7zAES
pwd1.txt *	1 KB	1 KB	文本文档		2016-10-05 23:58:	7C2DF918	LZMA2:18 7zAES
pwd2.txt *	1 KB	1 KB	文本文档		2016-10-05 10:45:	A58A1926	LZMA2:18 7zAES
pwd3.txt *	1 KB	1 KB	文本文档		2016-10-05 10:46:	4DAD5967	LZMA2:18 7zAES

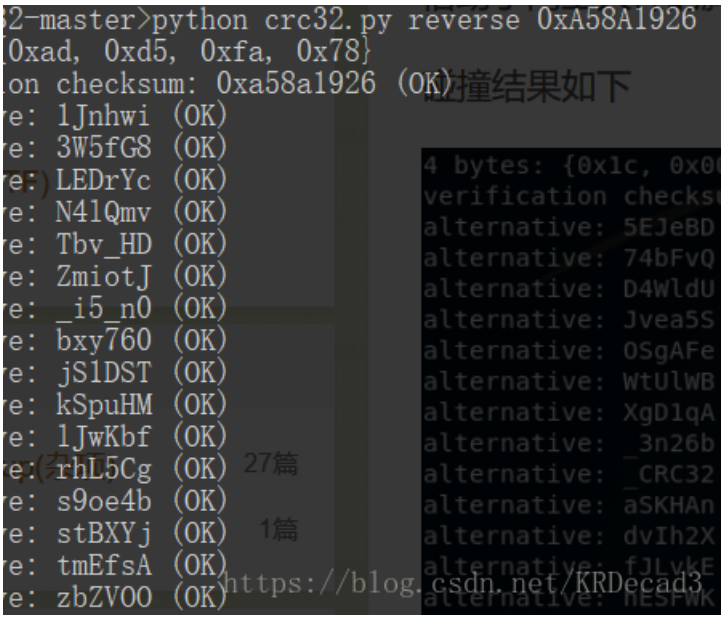
第一层，CRC32碰撞，参考大神的做法，用脚本碰撞，
https://github.com/theonlypwner/crc32
碰撞结果：

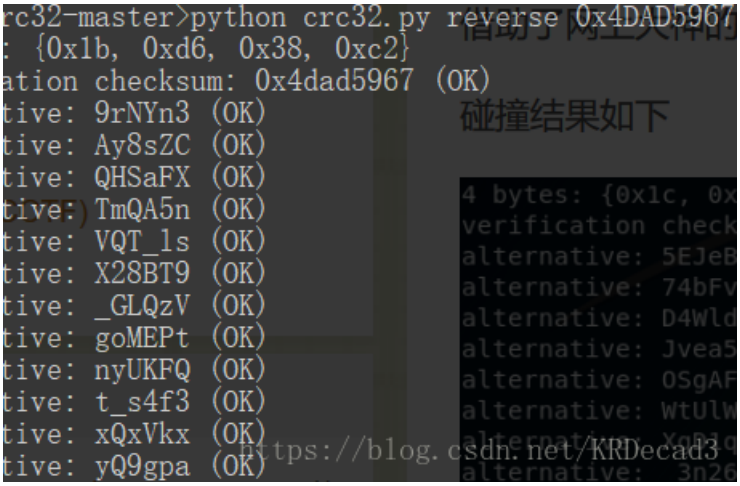
```
32-master>python crc32.py reverse 0x7C2DF918
{0x1c, 0x00, 0x1c, 0xa1}
ion checksum: 0x7c2df918 (OK)
ve: 5EJeBD (OK)
ve: 74bFvQ (OK)
ve: D4W1dU (OK)
```

借助了网上大神的

碰撞结果如下







找到每一次碰撞产生的看起来有意义的字符：“_CRC32”，“_i5_n0”，“t_s4f3”拼接起来就是压缩包的密码；
第二层：维吉尼亚密码，

你知道维吉尼亚密码吗？
我们给了keys.txt，唯一的密钥就在其中，那么解密ciphertext.txt里的密文吧！
解压密码就在明文里，祝你好运！

Do you know the Vigenére Ciphers?
We gave the keys.txt, Only have a key in it, So decrypts ciphertext.txt!
Unzip Password in plaintext, good luck to you!

这个没弄明白。。。

第三层：sha1碰撞

不完整的密码："*7*5-*4*3?" *代表可打印字符

不完整的sha1："619c20c*a4de755*9be9a8b*b7cbfa5*e8b4365*" *代表可打印字符

上脚本：

```
# -*- coding:utf-8 -*-

import hashlib
import string
import re

payload = string.printable

password = "%s7%s5-%s4%s3?"

sha1 = "619c20c.a4de755.9be9a8b.b7cbfa5.e8b4365."

for a in payload:
    for b in payload:
        for c in payload:
            for d in payload:
                pwd = password %(a,b,c,d)
                pwd = pwd.encode()
                if re.findall(sha1,hashlib.sha1(pwd).hexdigest()):
                    print (pwd)
                    break
```

第四层：MD5校验

安全客上有篇相关的文章http://bobao.360.cn/news/detail/768.html

程序下载下来，运行得到“Goodbye World :-(”

第五层：RSA

使用openssl导入公钥，查看模数n和指数e,

```
root@kali:~/桌面# openssl rsa -inform PEM -in rsa_public_key.pem -noout -modulus -tex
t -pubin
Public-Key: (1026 bit)
Modulus:
    02:8f:ff:9d:d3:e6:fe:97:81:64:9e:b7:fe:5e:93:
    03:cf:69:63:47:c4:11:0b:c4:ba:39:69:f0:b1:16:
    69:84:0c:51:d8:1a:68:42:b6:df:2b:09:0f:21:cd:
    76:d4:37:1a:8c:0e:47:04:8c:96:5e:ca:5b:46:91:
    3a:fb:b8:da:05:20:72:a0:56:6d:70:39:c6:18:ab:
    a9:06:57:59:b0:59:e2:9e:48:5d:c5:06:1a:16:ac:
    63:12:94:38:d9:35:4e:65:df:57:47:54:6b:85:db:
    3d:69:98:19:c4:b7:73:2d:f9:27:c7:08:4a:5d:52:
    d6:e6:d6:aa:c1:44:62:34:25
Exponent:
```

```
    01:f8:fb:a4:10:05:2d:f7:ed:a3:46:2f:1a:ac:d6:
    9e:40:76:04:33:ca:33:57:67:cd:73:05:a3:d0:90:
    80:5a:5f:d4:05:dd:6e:ea:70:e9:8f:0c:a1:e1:cf:
    25:47:48:67:1b:f0:c9:80:06:c2:0e:ee:1d:62:79:
    04:35:09:fe:7a:98:23:8b:43:91:60:a5:61:2d:a7:
    1e:90:45:14:e8:12:80:61:7e:30:7c:3c:d3:31:3f:
    a4:c6:fc:a3:31:59:d0:44:1f:bb:18:d8:3c:af:4b:
    d4:6f:6b:92:97:a8:0a:14:2d:d6:9b:f1:a3:57:cc:
    b5:e4:c2:00:b6:d9:0f:15:a3
Modulus=28FFF9DD3E6FE9781649EB7FE5E9303CF696347C4110BC4BA3969F0B11669840C51D81A6842B6
DF2B090F21CD76D4371A8C0E47048C965ECA5B46913AFBB8DA052072A0566D7039C618ABA9065759B059E
29E485DC5061A16AC63129438D9354E65DF5747546B85DB3D699819C4B7732DF927C7084A5D52D6E6D6AA
C144623425
```

可以看到指数（Exponent）很大，在RSA中如果n确定，e非常大，会导致d很小，从而出现维纳攻击，使用连分式（Continued fraction）去求得d。

维纳攻击的工具：

https://github.com/pablocelayes/rsa-wiener-attack

修改一下RSAwienerHacker.py

```
if __name__ == "__main__":
    #test_is_perfect_square()
    #print("--------------------------")
    n =
    0x28FFF9DD3E6FE9781649EB7FE5E9303CF696347C4110BC4BA3969F
    0B11669840C51D81A6842B6DF2B090F21CD76D4371A8C0E47048C965
    ECA5B46913AFBB8DA052072A0566D7039C618ABA9065759B059E29E4
    85DC5061A16AC63129438D9354E65DF5747546B85DB3D699819C4B77
    32DF927C7084A5D52D6E6D6AAC144623425
    e =
    0x01f8fba410052df7eda3462f1aacd69e40760433ca335767cd7305
    a3d090805a5fd405dd6eea70e98f0ca1e1cf254748671bf0c98006c2
    0eee1d6279043509fe7a98238b439160a5612da71e904514e8128061
    7e307c3cd3313fa4c6fca33159d0441fbb18d83caf4bd46f6b9297a8
    0a142dd69bf1a357ccb5e4c200b6d90f15a3
    d = hack_RSA(e, n)
    print "d=",d
```


求得d，

再使用rsatool生成私钥文件，得到rsa_private_key.pem，

再用openssl对flag.enx解密

这里有一个比较详细的wp:

https://mp.weixin.qq.com/s/5_gxomJYbTjXISLhGoMoxg