2020祥云杯网络安全大赛 MISC Writeup

原创

末 初 于 2020-11-23 17:25:14 发布 4312 收藏 45

分类专栏： CTF_MISC_Writeup

本文链接：https://blog.csdn.net/mochu7777777/article/details/109913252

版权

CTF_MISC_Writeup 专栏收录该内容

246 篇文章 46 订阅

订阅专栏

文章目录

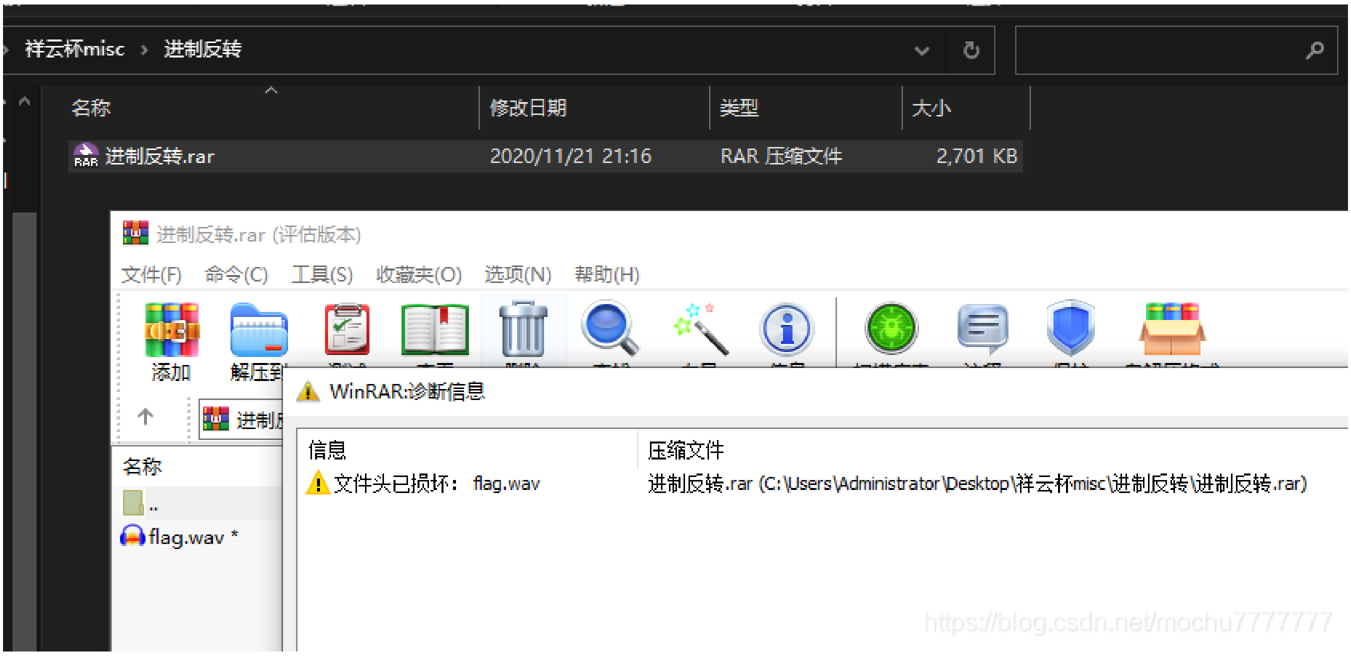
签到

```
PS C:\Users\Administrator> php -r "var_dump(base64_decode('ZmxhZ3txcV9ncm91cF84MjY1NjYwNDB9'));"
string(24) "flag{qq_group_826566040}"
```

进制反转

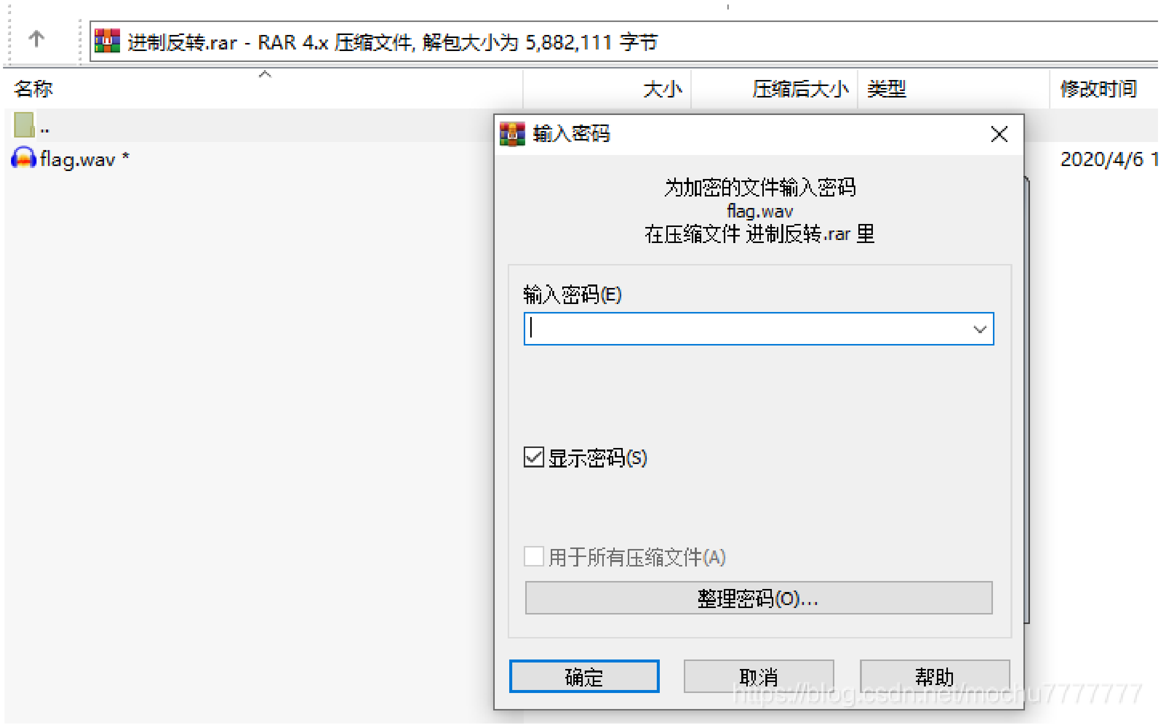
题目描述：

```
电脑中到底使用的是什么进制呢？真是麻烦，有时候还是手机好用。结果用flag{}包住，并且全为大写
```



WinRAR:诊断信息

信息	压缩文件
文件头已损坏： flag.wav	进制反转.rar (C:\Users\Administrator\Desktop\祥云杯misc\进制反转\进制反转.rar)


进制反转.rar - RAR 4.x 压缩文件, 解包大小为 5,882,111 字节

输入密码

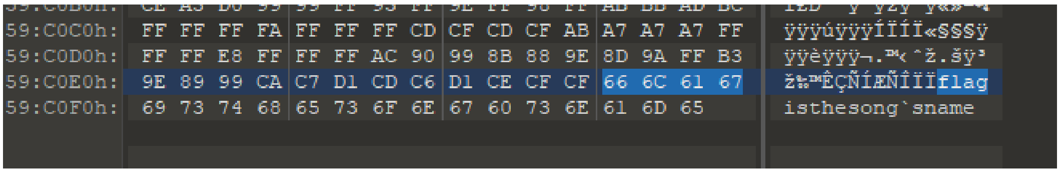
为加密的文件输入密码
flag.wav
在压缩文件 进制反转.rar 里

输入密码(E)

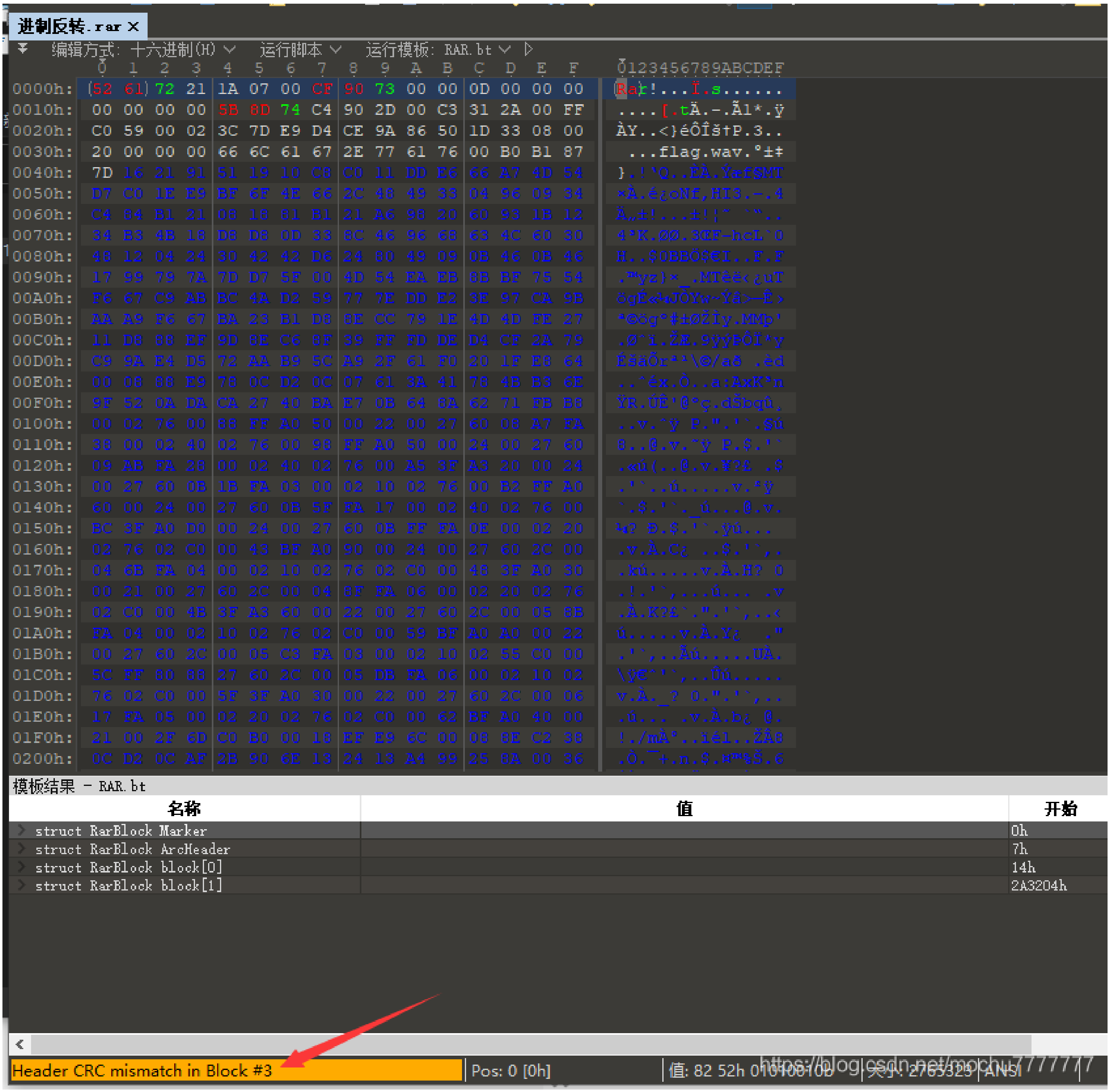
☑ 显示密码(S)

□ 用于所有压缩文件(A)

`WinRAR`打开显示`文件头损坏`，其次有加密，猜测`RAR伪加密`，使用`010 Editor`打开


```
59:C0C0h: FF FF FF FA FF FF FF CD CF CD CF AB A7 A7 A7 FF  ÿÿÿúÿÿÿÍÏÍÏ«§§§ÿ
59:C0D0h: FF FF E8 FF FF FF AC 90 99 8B 88 9E 8D 9A FF B3  ÿÿèÿÿÿ¬.™‹ˆž.šÿ³
59:C0E0h: 9E 89 99 CA C7 D1 CD C6 D1 CE CF CF 66 6C 61 67  ž‰™ÊÇÑÍÆÑÎÏÏflag
59:C0F0h: 69 73 74 68 65 73 6F 6E 67 60 73 6E 61 6D 65     isthesong`sname
```

文件结尾发现提示：`flag is the song's name`



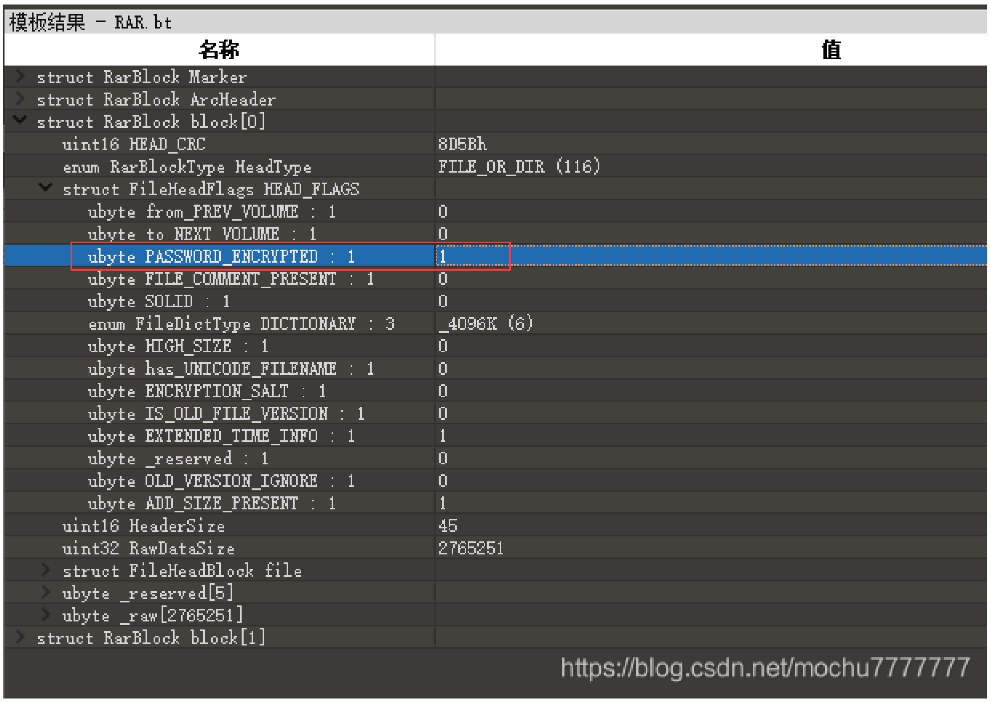
```
        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
0000h: 52 61 72 21 1A 07 00 CF 90 73 00 00 0D 00 00 00   Rar!...Ï.s......
0010h: 00 00 00 00 5B 8D 74 C4 90 2D 00 C3 31 2A 00 FF   ....[.tÄ.-.Ã1*.ÿ
0020h: C0 59 00 02 3C 7D E9 D4 CE 9A 86 50 1D 33 08 00   ÀY..<}éÔÎš†P.3..
0030h: 20 00 00 00 66 6C 61 67 2E 77 61 76 00 B0 B1 87    ...flag.wav.°±‡
0040h: 7D 16 21 91 51 19 10 C8 C0 11 DD E6 66 A7 4D 54   }.!'Q..ÈÀ.Ýæf§MT
0050h: D7 C0 1E E9 BF 6F 4E 66 2C 48 49 33 04 96 09 34   ×À.é¿oNf,HI3.–.4
0060h: C4 84 B1 21 08 18 81 B1 21 A6 98 20 60 93 1B 12   Ä„±!...±!¦˜ `"..
0070h: 34 B3 4B 18 D8 D8 0D 33 8C 46 96 68 63 4C 60 30   4³K.ØØ.3ŒF–hcL`0
0080h: 48 12 04 24 30 42 42 D6 24 80 49 09 0B 46 0B 46   H..$0BBÖ$€I..F.F
0090h: 17 99 79 7A 7D D7 5F 00 4D 54 EA EB 8B BF 75 54   .™yz}×_.MTêë‹¿uT
00A0h: F6 67 C9 AB BC 4A D2 59 77 7E DD E2 3E 97 CA 9B   ögÉ«¼JÒYw~Ýâ>—Ê›
00B0h: AA A9 F6 67 BA 23 B1 D8 8E CC 79 1E 4D 4D FE 27   ª©ögº#±ØŽÌy.MMþ'
00C0h: 11 D8 88 EF 9D 8E C6 8F 39 FF FD DE D4 CF 2A 79   .Øˆï.ŽÆ.9ÿýÞÔÏ*y
00D0h: C9 9A E4 D5 72 AA B9 5C A9 2F 61 F0 20 1F E8 64   ÉšäÕrª¹\©/að .èd
00E0h: 00 08 88 E9 78 0C D2 0C 07 61 3A 41 78 4B B3 6E   ..ˆéx.Ò..a:AxK³n
00F0h: 9F 52 0A DA CA 27 40 BA E7 0B 64 8A 62 71 FB B8   ŸR.ÚÊ'@ºç.dŠbqû¸
0100h: 00 02 76 00 88 FF A0 50 00 22 00 27 60 08 A7 FA   ..v.ˆÿ P.".'`.§ú
0110h: 38 00 02 40 02 76 00 98 FF A0 50 00 24 00 27 60   8..@.v.˜ÿ P.$.'`
0120h: 09 AB FA 28 00 02 40 02 76 00 A5 3F A3 20 00 24   .«ú(..@.v.¥?£ .$
0130h: 00 27 60 0B 1B FA 03 00 02 10 02 76 00 B2 FF A0   .'`..ú.....v.²ÿ 
0140h: 60 00 24 00 27 60 0B 5F FA 17 00 02 40 02 76 00   `.$.'`._ú...@.v.
0150h: BC 3F A0 D0 00 24 00 27 60 0B FF FA 0E 00 02 20   ¼? Ð.$.'`.ÿú... 
0160h: 02 76 02 C0 00 43 BF A0 90 00 24 00 27 60 2C 00   .v.À.C¿ ..$.'`,.
0170h: 04 6B FA 04 00 02 10 02 76 02 C0 00 48 3F A0 30   .kú.....v.À.H? 0
0180h: 00 21 00 27 60 2C 00 04 8F FA 06 00 02 20 02 76   .!.'`,...ú... .v
0190h: 02 C0 00 4B 3F A3 60 00 22 00 27 60 2C 00 05 8B   .À.K?£`.".'`,..‹
01A0h: FA 04 00 02 10 02 76 02 C0 00 59 BF A0 A0 00 22   ú.....v.À.Y¿  ."
01B0h: 00 27 60 2C 00 05 C3 FA 03 00 02 10 02 55 C0 00   .'`,..Ãú.....UÀ.
01C0h: 5C FF 80 88 27 60 2C 00 05 DB FA 06 00 02 10 02   \ÿ€ˆ'`,..Ûú.....
01D0h: 76 02 C0 00 5F 3F A0 30 00 22 00 27 60 2C 00 06   v.À._? 0.".'`,..
01E0h: 17 FA 05 00 02 20 02 76 02 C0 00 62 BF A0 40 00   .ú... .v.À.b¿ @.
01F0h: 21 00 2F 6D C0 B0 00 18 EF E9 6C 00 08 8E C2 38   !./mÀ°..ïél..ŽÂ8
0200h: 0C D2 0C AF 2B 90 6E 13 24 13 A4 99 25 8A 00 36   .Ò.¯+.n.$.¤™%Š.6
```

模板结果 - RAR.bt

名称	值	开始
struct RarBlock Marker		0h
struct RarBlock ArcHeader		7h
struct RarBlock block[0]		14h
struct RarBlock block[1]		2A3204h

Header CRC mismatch in Block #3 | Pos: 0 [0h] | 值: 82 52h 01010010b | 大小: 2765325 | ANSI

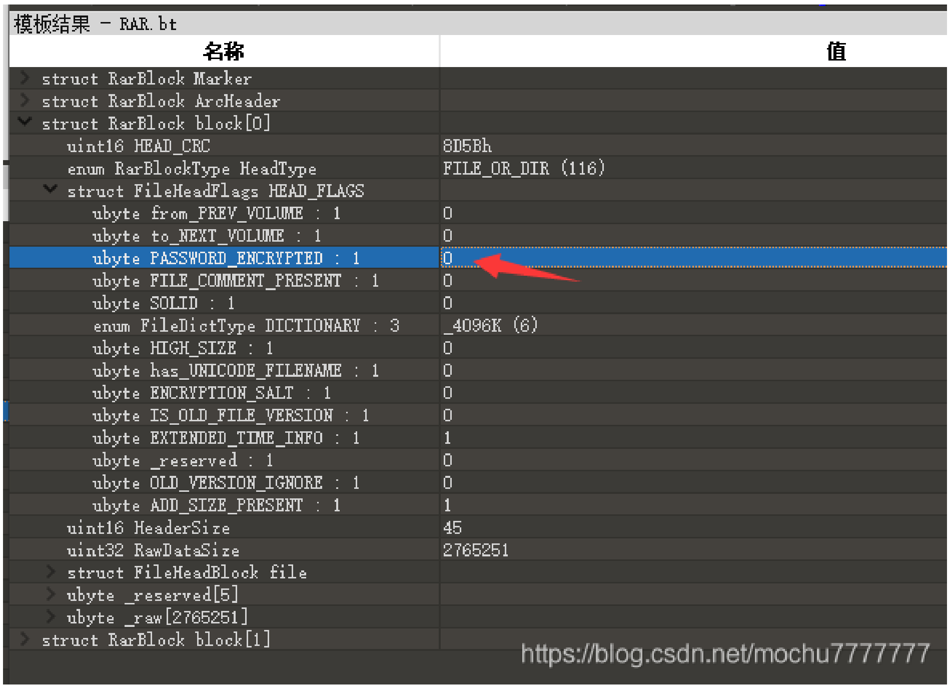
接着找到第三块`struct RarBlock block[0]`下的`struct FileHeadFlags HEAD_FLAGS`



模板结果 - RAR.bt

名称	值
struct RarBlock Marker	
struct RarBlock ArcHeader	
struct RarBlock block[0]	
uint16 HEAD_CRC	8D5Bh
enum RarBlockType HeadType	FILE_OR_DIR (116)
struct FileHeadFlags HEAD_FLAGS	
ubyte from_PREV_VOLUME : 1	0
ubyte to_NEXT_VOLUME : 1	0
ubyte PASSWORD_ENCRYPTED : 1	1
ubyte FILE_COMMENT_PRESENT : 1	0
ubyte SOLID : 1	0
enum FileDictType DICTIONARY : 3	_4096K (6)
ubyte HIGH_SIZE : 1	0
ubyte has_UNICODE_FILENAME : 1	0
ubyte ENCRYPTION_SALT : 1	0
ubyte IS_OLD_FILE_VERSION : 1	0
ubyte EXTENDED_TIME_INFO : 1	1
ubyte _reserved : 1	0
ubyte OLD_VERSION_IGNORE : 1	0
ubyte ADD_SIZE_PRESENT : 1	1
uint16 HeaderSize	45
uint32 RawDataSize	2765251
struct FileHeadBlock file	
ubyte _reserved[5]	
ubyte _raw[2765251]	
struct RarBlock block[1]	

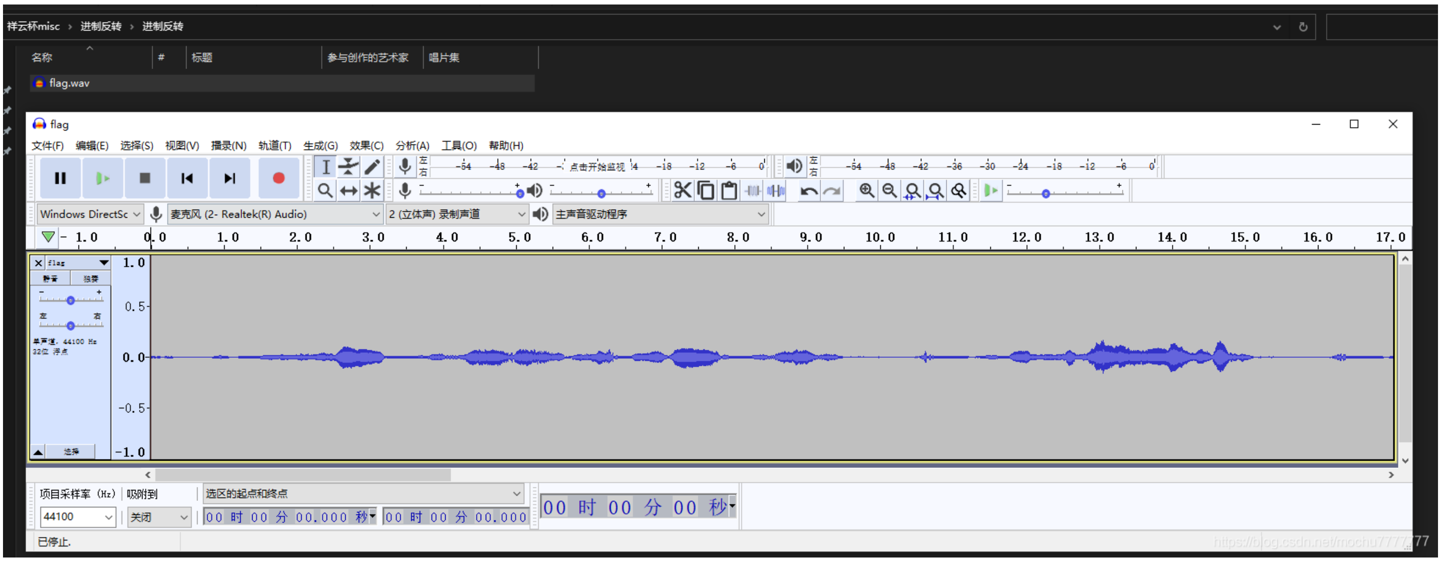
修改`ubyte PASSWORD_ENCRYPTED`的值为`0`



模板结果 - RAR.bt

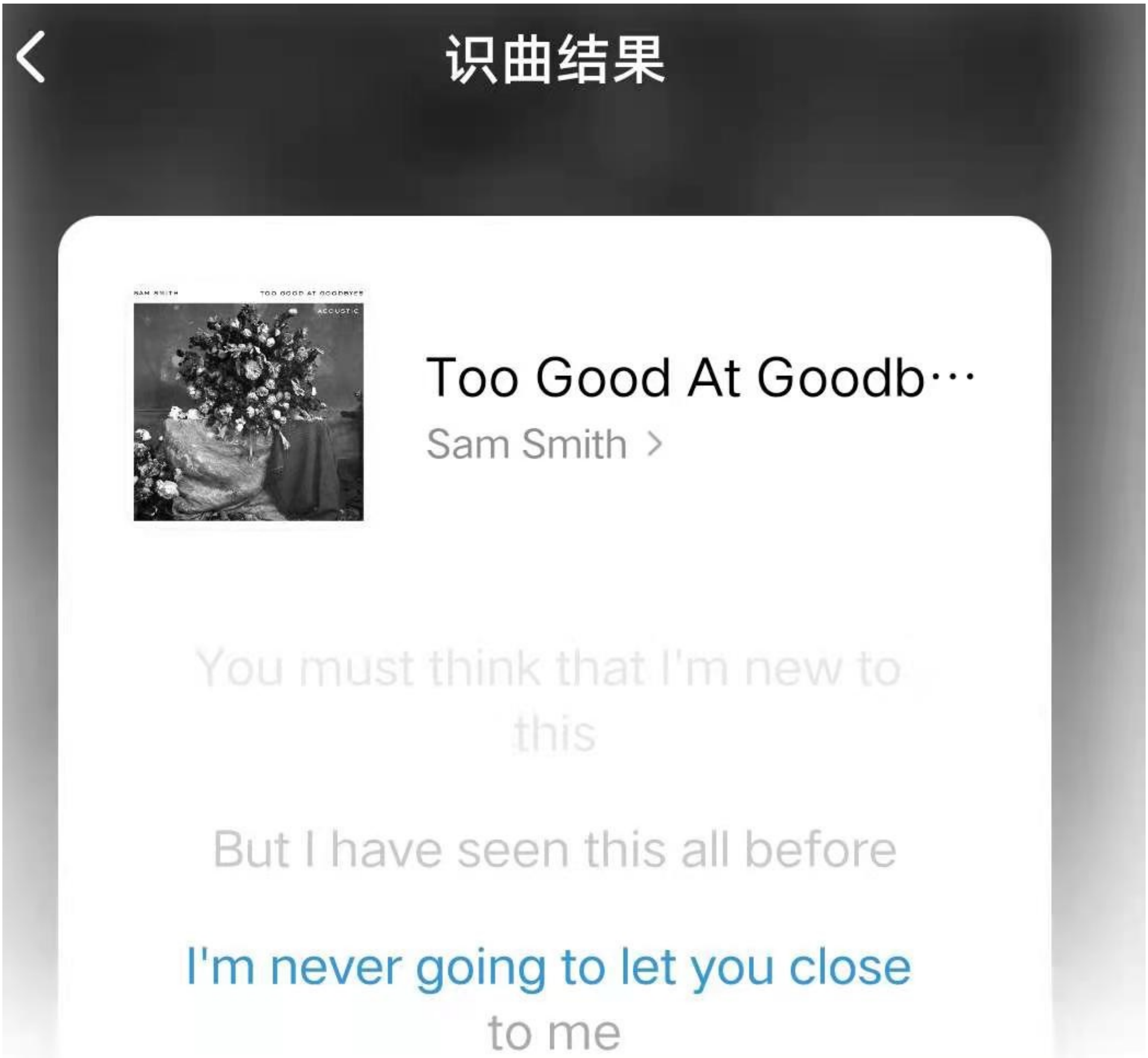
名称	值
struct RarBlock Marker	
struct RarBlock ArcHeader	
struct RarBlock block[0]	
uint16 HEAD_CRC	8D5Bh
enum RarBlockType HeadType	FILE_OR_DIR (116)
struct FileHeadFlags HEAD_FLAGS	
ubyte from_PREV_VOLUME : 1	0
ubyte to_NEXT_VOLUME : 1	0
ubyte PASSWORD_ENCRYPTED : 1	0
ubyte FILE_COMMENT_PRESENT : 1	0
ubyte SOLID : 1	0
enum FileDictType DICTIONARY : 3	_4096K (6)
ubyte HIGH_SIZE : 1	0
ubyte has_UNICODE_FILENAME : 1	0
ubyte ENCRYPTION_SALT : 1	0
ubyte IS_OLD_FILE_VERSION : 1	0
ubyte EXTENDED_TIME_INFO : 1	1
ubyte _reserved : 1	0
ubyte OLD_VERSION_IGNORE : 1	0
ubyte ADD_SIZE_PRESENT : 1	1
uint16 HeaderSize	45
uint32 RawDataSize	2765251
struct FileHeadBlock file	
ubyte _reserved[5]	
ubyte _raw[2765251]	
struct RarBlock block[1]	

解压得到`flag.wav`，无法使用`Audacity`打开，就通过导入`文件->导入->原始数据`



听着很明显是歌声但是却是倒放，Ctrl+A全选，点击效果 > 反向(时间)，然后再效果 > 改变速率，调节到一个正常歌曲的播放速度，然后经过降噪，消除咔嚓声等一系列操作，最后听歌识别

先推个在线识别歌曲网站：https://www.acrcloud.com/identify-songs-music-recognition-online/



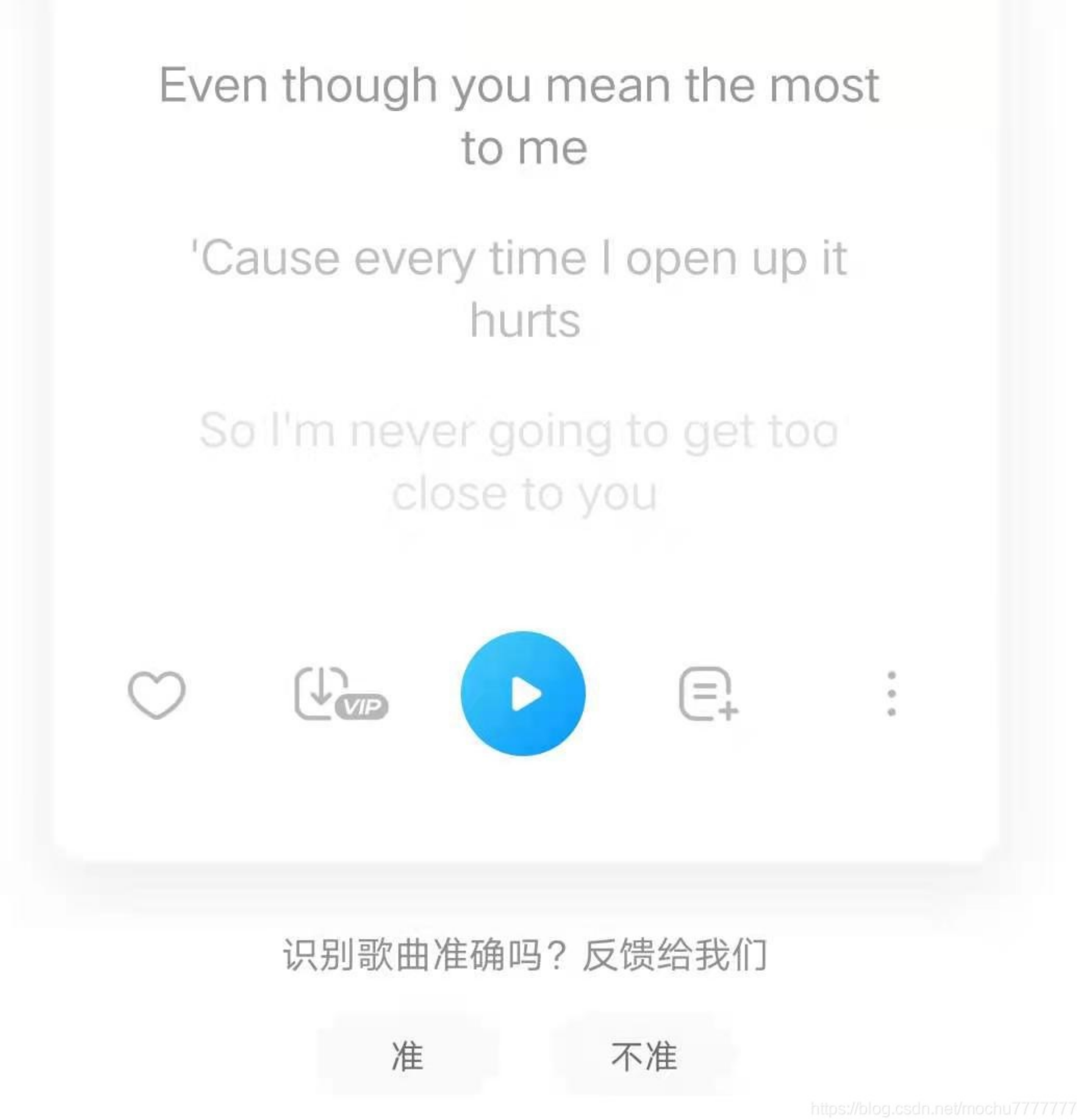
Too Good At Goodb···

Sam Smith >

You must think that I'm new to this

But I have seen this all before

I'm never going to let you close to me



Even though you mean the most to me

'Cause every time I open up it hurts

So I'm never going to get too close to you

识别歌曲准确吗？反馈给我们

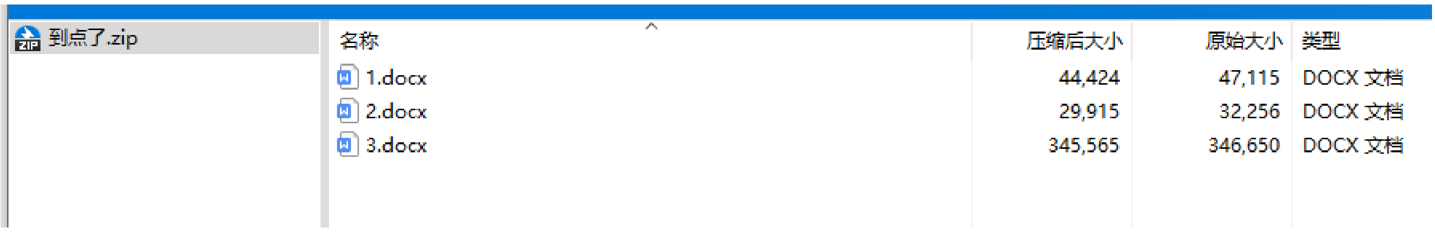
听歌识曲识别不出来，就听歌词找吧，也挺快的，考验听力水平
歌名：《Too Good At Goodbyes》

```
flag{TOOGOODATGOODBYES}
```

到点了

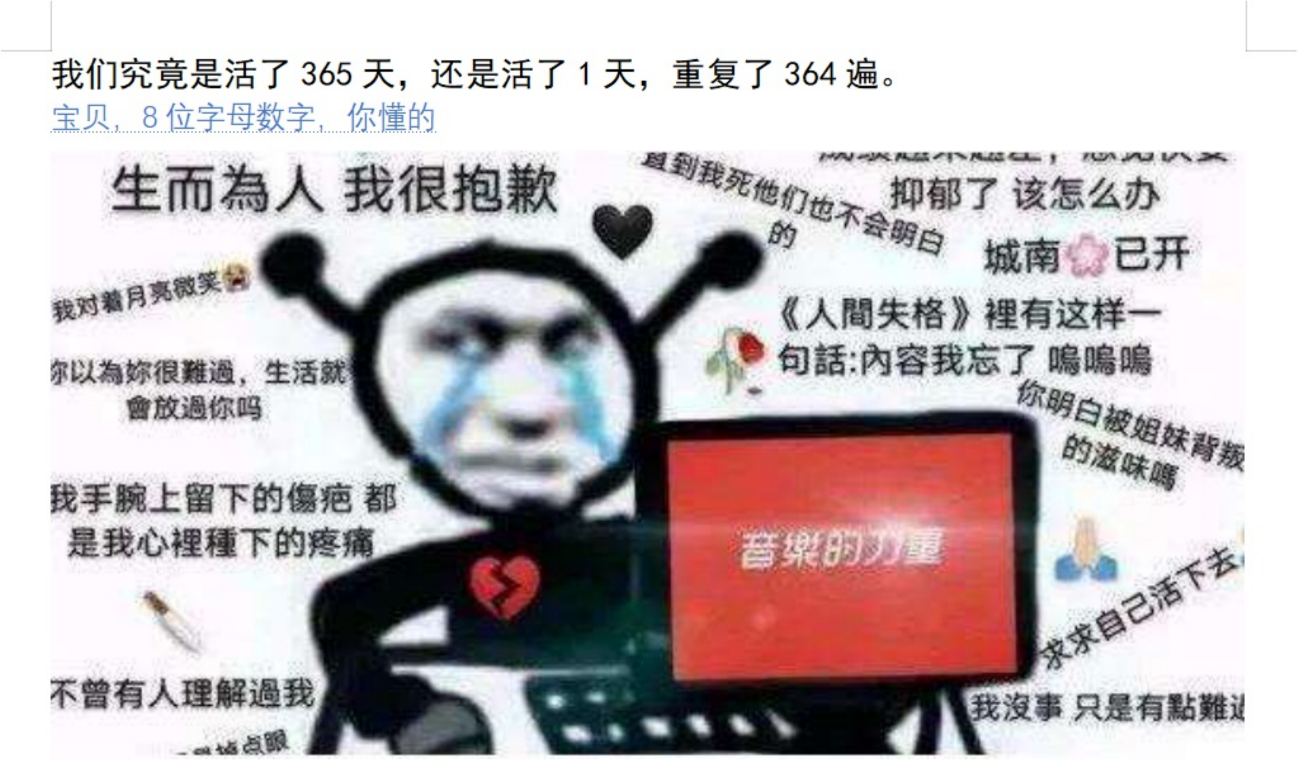
题目描述：

```
我那么多遗憾，那么多期盼，你知道吗（下雨熊猫头
```



名称	压缩后大小	原始大小	类型
1.docx	44,424	47,115	DOCX 文档
2.docx	29,915	32,256	DOCX 文档
3.docx	345,565	346,650	DOCX 文档

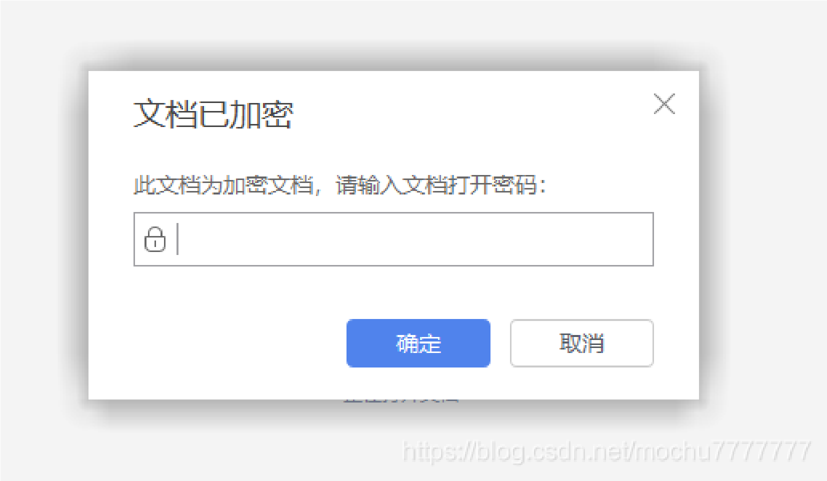
`1.docx`打开，勾选隐藏文字



我们究竟是活了 365 天，还是活了 1 天，重复了 364 遍。

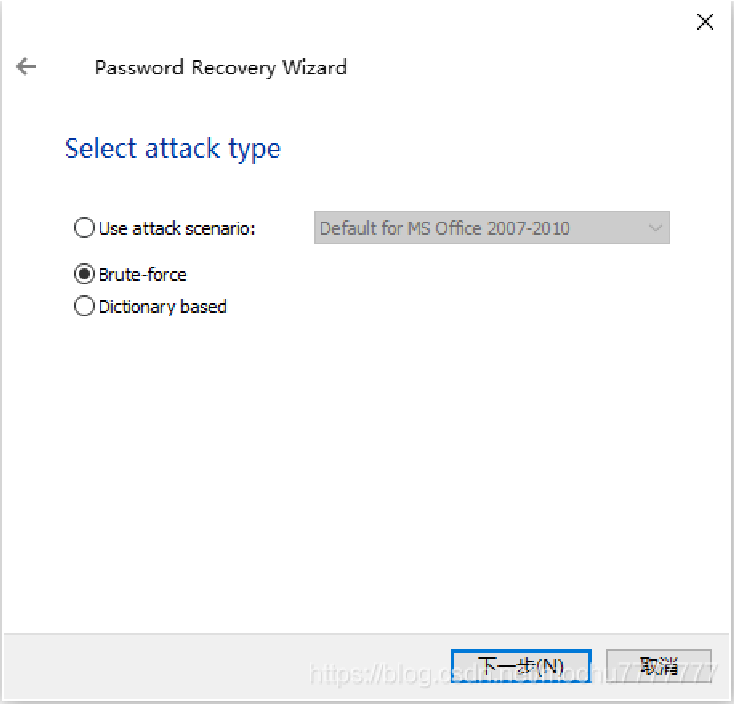
宝贝，8 位字母数字，你懂的

`2.docx`有加密，根据`1.docx`提供的提示，使用`Accent OFFICE Password Recovery`爆破密码



文档已加密

此文档为加密文档，请输入文档打开密码：



Password Recovery Wizard

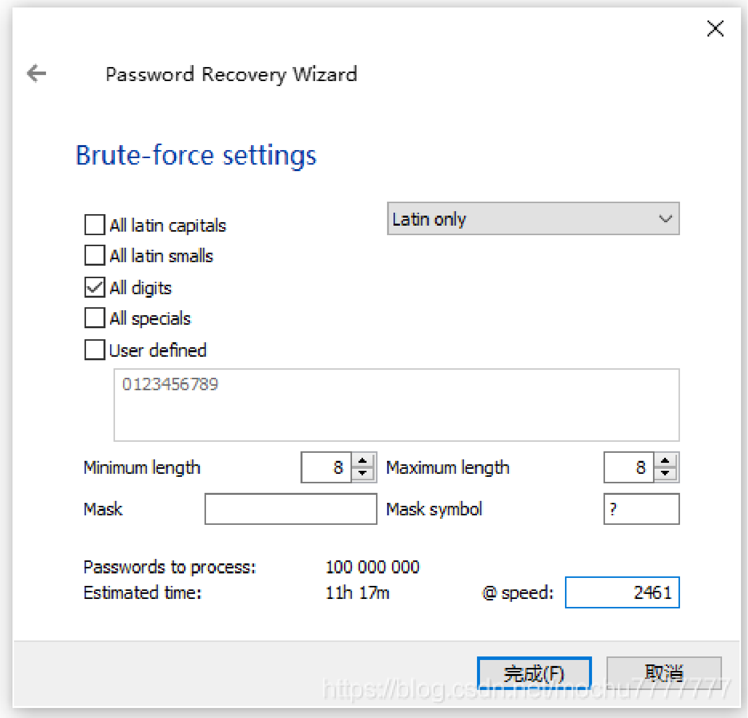
Select attack type

○ Use attack scenario: Default for MS Office 2007-2010

● Brute-force

○ Dictionary based

先尝试爆破8位纯数字，毕竟8位字母数字就太多了，还不知道分不分大小写



Password Recovery Wizard

Brute-force settings

□ All latin capitals
□ All latin smalls
☑ All digits
□ All specials
□ User defined

Latin only

0123456789

Minimum length 8 Maximum length 8

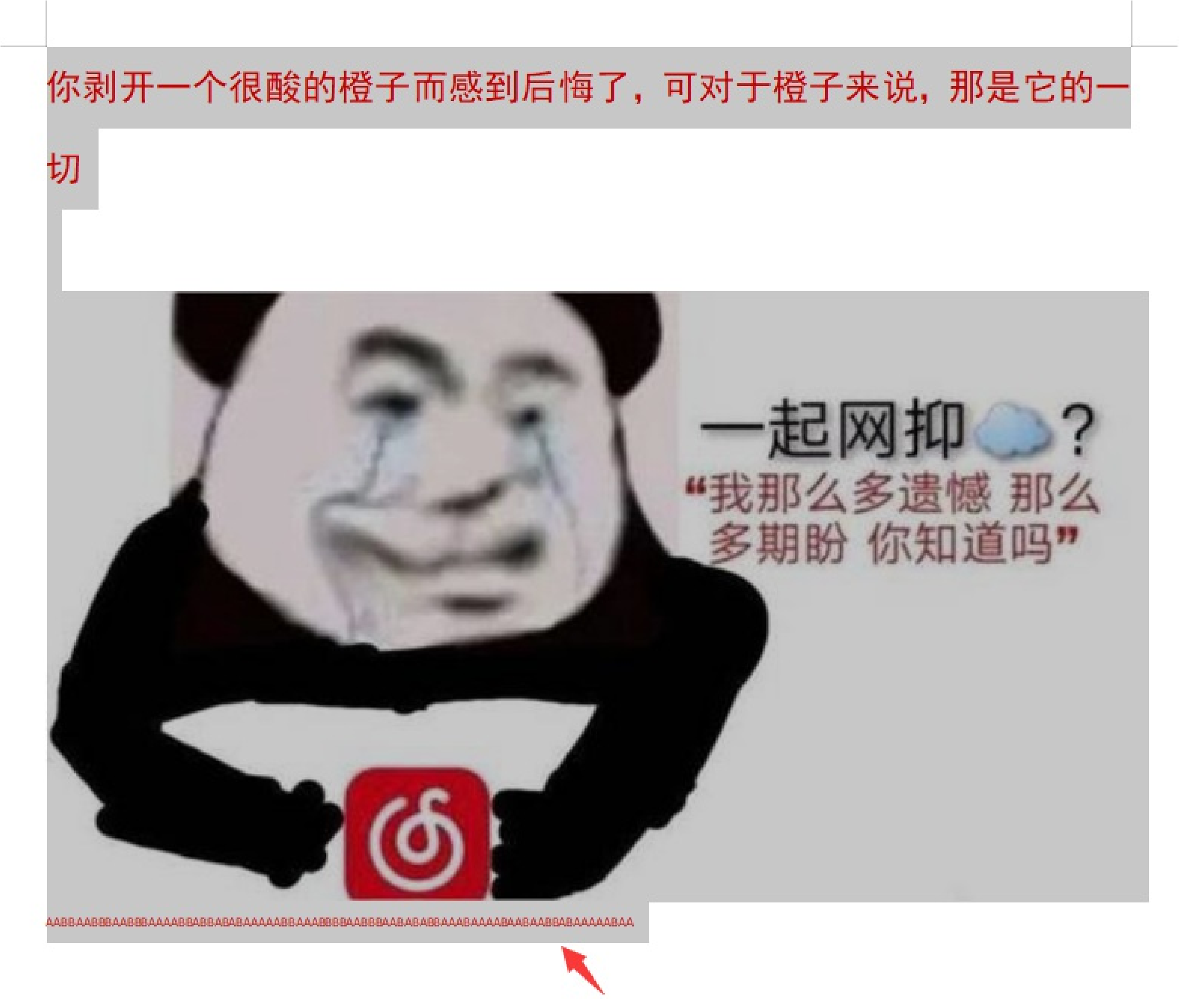
Mask Mask symbol ?

Passwords to process: 100 000 000

Estimated time: 11h 17m @ speed: 2461

爆破过程就不看了，时间太长了，直接贴结果，密码为：`20201024`

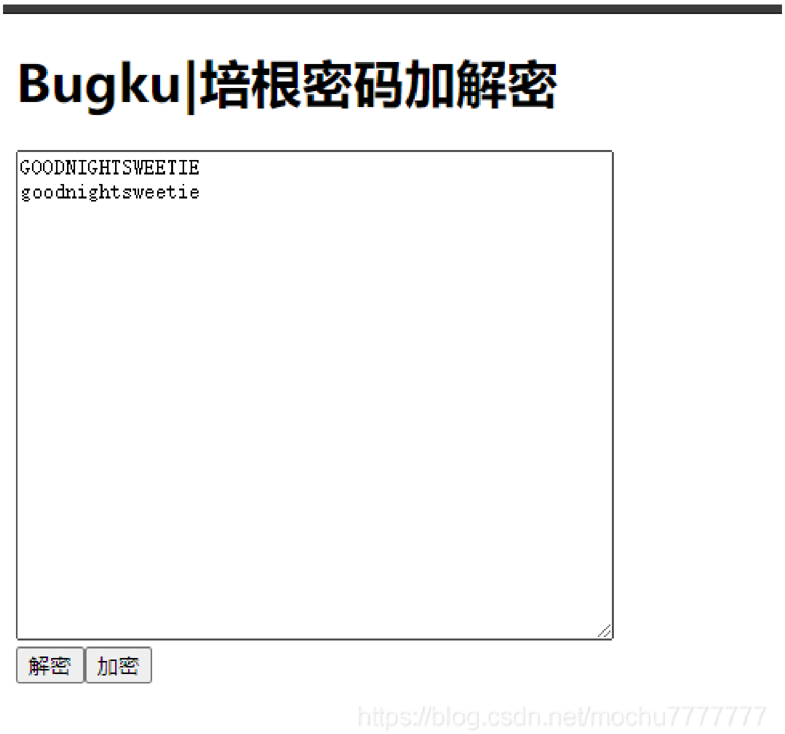
解开`2.docx`，全选标红，发现有一串`AB`字符，很明显应该是`培根密码`



你剥开一个很酸的橙子而感到后悔了，可对于橙子来说，那是它的一切

```
AABBAABBBAABBBAAAABBABBABABAAAAABBAAABBBBAABBBAABABABBAAABAAAABAABAABBABAAAAAABAA
```

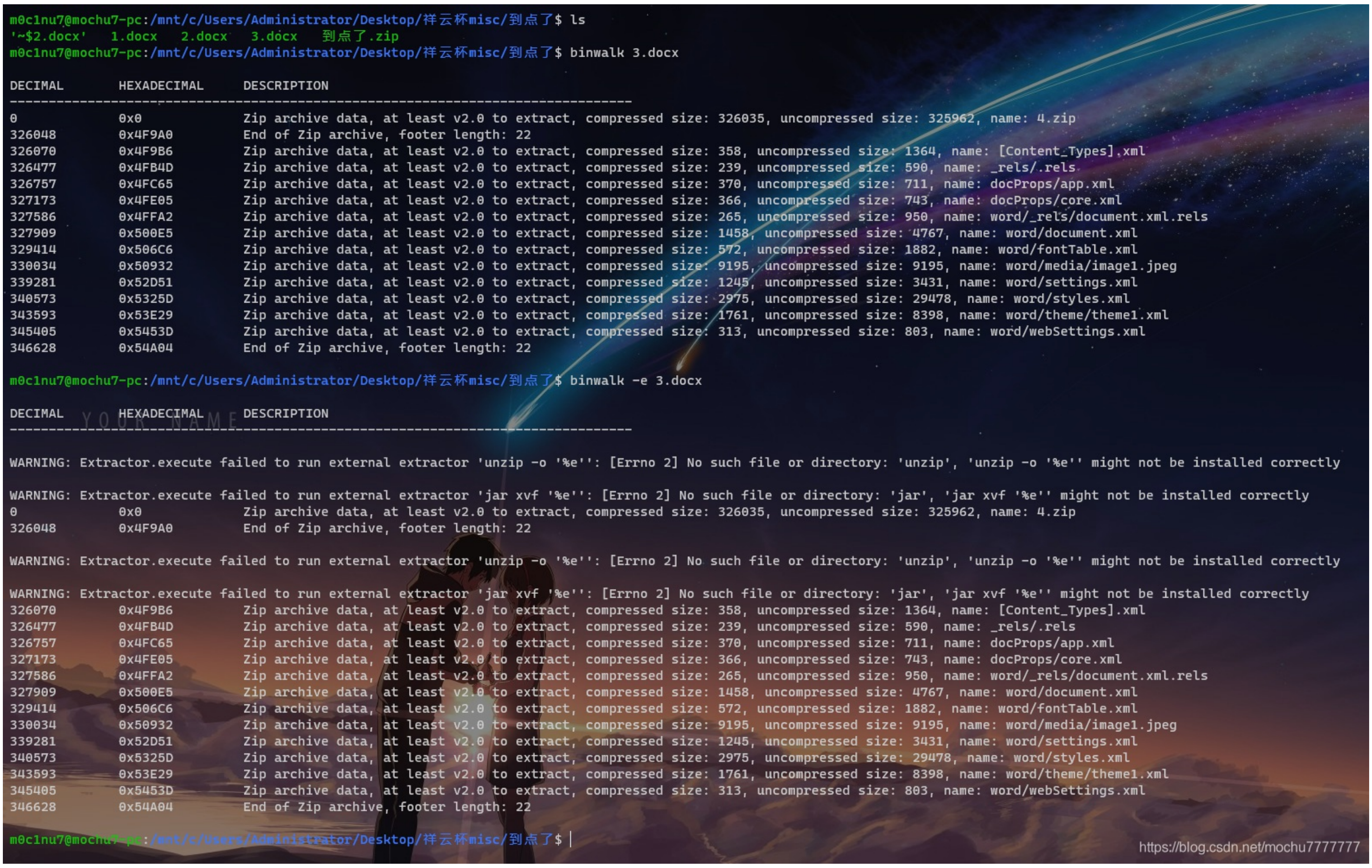
培根在线解码：https://tool.bugku.com/peigen/



Bugku|培根密码加解密

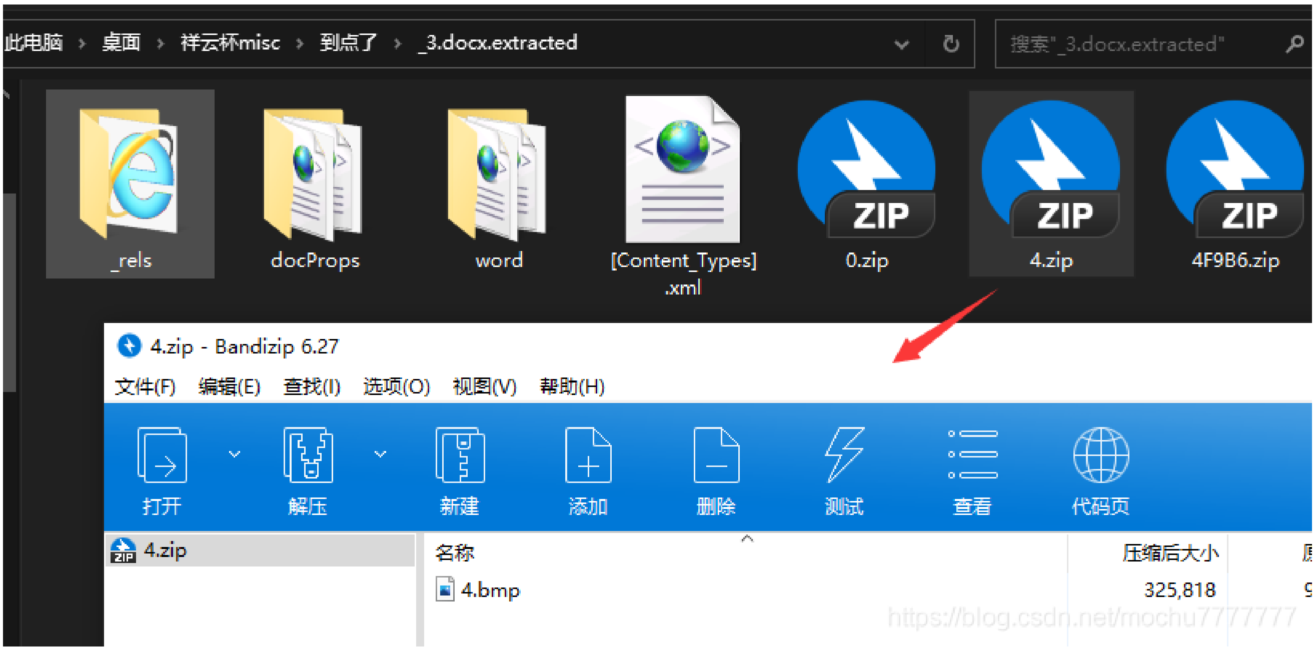
GOODNIGHTSWEETIE
goodnightsweetie

```
GOODNIGHTSWEETIE
goodnightsweetie
```



```
m0c1nu7@mochu7-pc:/mnt/c/Users/Administrator/Desktop/祥云杯misc/到点了$ ls
'~$2.docx'   1.docx   2.docx   3.docx   到点了.zip
m0c1nu7@mochu7-pc:/mnt/c/Users/Administrator/Desktop/祥云杯misc/到点了$ binwalk 3.docx

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             Zip archive data, at least v2.0 to extract, compressed size: 326035, uncompressed size: 325962, name: 4.zip
326048        0x4F9A0         End of Zip archive, footer length: 22
326070        0x4F9B6         Zip archive data, at least v2.0 to extract, compressed size: 358, uncompressed size: 1364, name: [Content_Types].xml
326477        0x4FB4D         Zip archive data, at least v2.0 to extract, compressed size: 239, uncompressed size: 590, name: _rels/.rels
326757        0x4FC65         Zip archive data, at least v2.0 to extract, compressed size: 370, uncompressed size: 711, name: docProps/app.xml
327173        0x4FE05         Zip archive data, at least v2.0 to extract, compressed size: 366, uncompressed size: 743, name: docProps/core.xml
327586        0x4FFA2         Zip archive data, at least v2.0 to extract, compressed size: 265, uncompressed size: 950, name: word/_rels/document.xml.rels
327909        0x500E5         Zip archive data, at least v2.0 to extract, compressed size: 1458, uncompressed size: 4767, name: word/document.xml
329414        0x506C6         Zip archive data, at least v2.0 to extract, compressed size: 572, uncompressed size: 1882, name: word/fontTable.xml
330034        0x50932         Zip archive data, at least v2.0 to extract, compressed size: 9195, uncompressed size: 9195, name: word/media/image1.jpeg
339281        0x52D51         Zip archive data, at least v2.0 to extract, compressed size: 1245, uncompressed size: 3431, name: word/settings.xml
340573        0x5325D         Zip archive data, at least v2.0 to extract, compressed size: 2975, uncompressed size: 29478, name: word/styles.xml
343593        0x53E29         Zip archive data, at least v2.0 to extract, compressed size: 1761, uncompressed size: 8398, name: word/theme/theme1.xml
345405        0x5453D         Zip archive data, at least v2.0 to extract, compressed size: 313, uncompressed size: 803, name: word/webSettings.xml
346628        0x54A04         End of Zip archive, footer length: 22

m0c1nu7@mochu7-pc:/mnt/c/Users/Administrator/Desktop/祥云杯misc/到点了$ binwalk -e 3.docx

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------

WARNING: Extractor.execute failed to run external extractor 'unzip -o '%e'': [Errno 2] No such file or directory: 'unzip', 'unzip -o '%e'' might not be installed correctly

WARNING: Extractor.execute failed to run external extractor 'jar xvf '%e'': [Errno 2] No such file or directory: 'jar', 'jar xvf '%e'' might not be installed correctly
0             0x0             Zip archive data, at least v2.0 to extract, compressed size: 326035, uncompressed size: 325962, name: 4.zip
326048        0x4F9A0         End of Zip archive, footer length: 22

WARNING: Extractor.execute failed to run external extractor 'unzip -o '%e'': [Errno 2] No such file or directory: 'unzip', 'unzip -o '%e'' might not be installed correctly

WARNING: Extractor.execute failed to run external extractor 'jar xvf '%e'': [Errno 2] No such file or directory: 'jar', 'jar xvf '%e'' might not be installed correctly
326070        0x4F9B6         Zip archive data, at least v2.0 to extract, compressed size: 358, uncompressed size: 1364, name: [Content_Types].xml
326477        0x4FB4D         Zip archive data, at least v2.0 to extract, compressed size: 239, uncompressed size: 590, name: _rels/.rels
326757        0x4FC65         Zip archive data, at least v2.0 to extract, compressed size: 370, uncompressed size: 711, name: docProps/app.xml
327173        0x4FE05         Zip archive data, at least v2.0 to extract, compressed size: 366, uncompressed size: 743, name: docProps/core.xml
327586        0x4FFA2         Zip archive data, at least v2.0 to extract, compressed size: 265, uncompressed size: 950, name: word/_rels/document.xml.rels
327909        0x500E5         Zip archive data, at least v2.0 to extract, compressed size: 1458, uncompressed size: 4767, name: word/document.xml
329414        0x506C6         Zip archive data, at least v2.0 to extract, compressed size: 572, uncompressed size: 1882, name: word/fontTable.xml
330034        0x50932         Zip archive data, at least v2.0 to extract, compressed size: 9195, uncompressed size: 9195, name: word/media/image1.jpeg
339281        0x52D51         Zip archive data, at least v2.0 to extract, compressed size: 1245, uncompressed size: 3431, name: word/settings.xml
340573        0x5325D         Zip archive data, at least v2.0 to extract, compressed size: 2975, uncompressed size: 29478, name: word/styles.xml
343593        0x53E29         Zip archive data, at least v2.0 to extract, compressed size: 1761, uncompressed size: 8398, name: word/theme/theme1.xml
345405        0x5453D         Zip archive data, at least v2.0 to extract, compressed size: 313, uncompressed size: 803, name: word/webSettings.xml
346628        0x54A04         End of Zip archive, footer length: 22

m0c1nu7@mochu7-pc:/mnt/c/Users/Administrator/Desktop/祥云杯misc/到点了$ |
```

binwalk分离`3.docx`，得到一个`4.zip`，里面有一张`4.bmp`

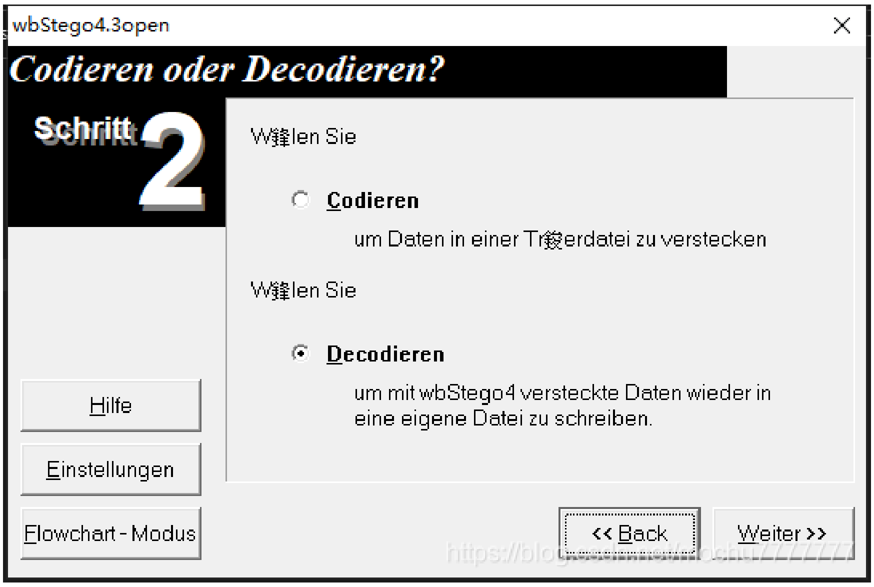

此电脑 › 桌面 › 祥云杯misc › 到点了 › _3.docx.extracted

名称	压缩后大小
4.bmp	325,818

`4.bmp`

雨淋湿了天空
毁得很讲究

`bmp隐写`，有密码，试了不是LSB，尝试使用`wbs43open`



Codieren oder Decodieren?

W鋒len Sie

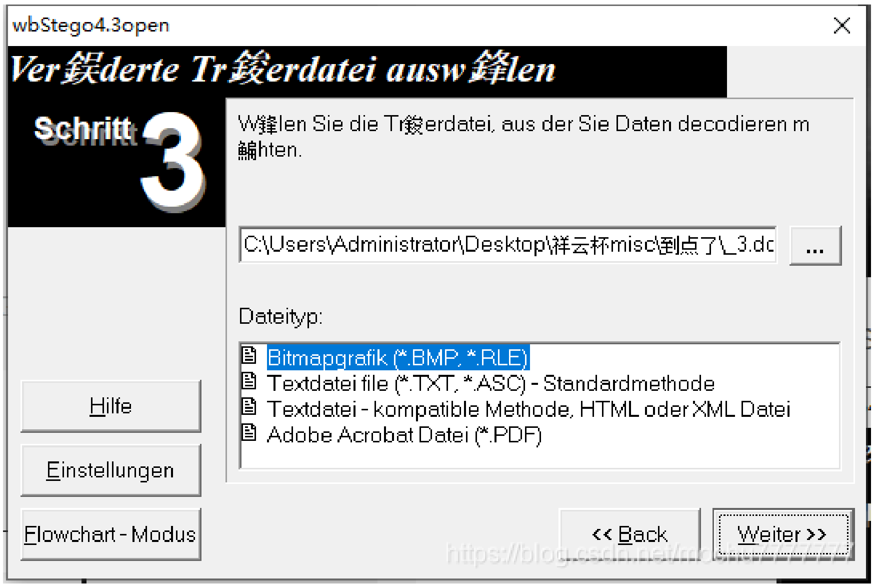
○ **Codieren**

um Daten in einer Tr鋑erdatei zu verstecken

W鋒len Sie

● **Decodieren**

um mit wbStego4 versteckte Daten wieder in eine eigene Datei zu schreiben.



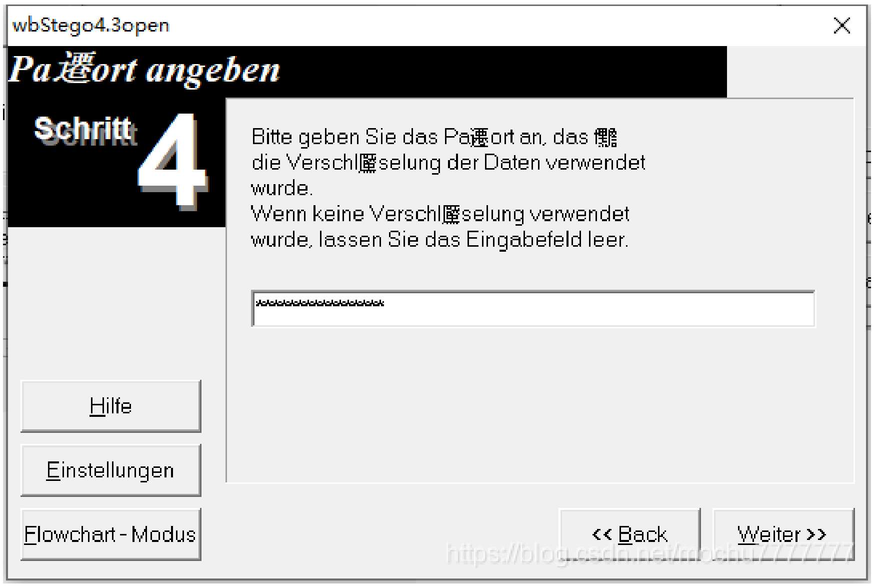
Ver鋘derte Tr鋑erdatei ausw鋒len

W鋒len Sie die Tr鋑erdatei, aus der Sie Daten decodieren m鰄hten.

C:\Users\Administrator\Desktop\祥云杯misc\到点了_3.dc

Dateityp:

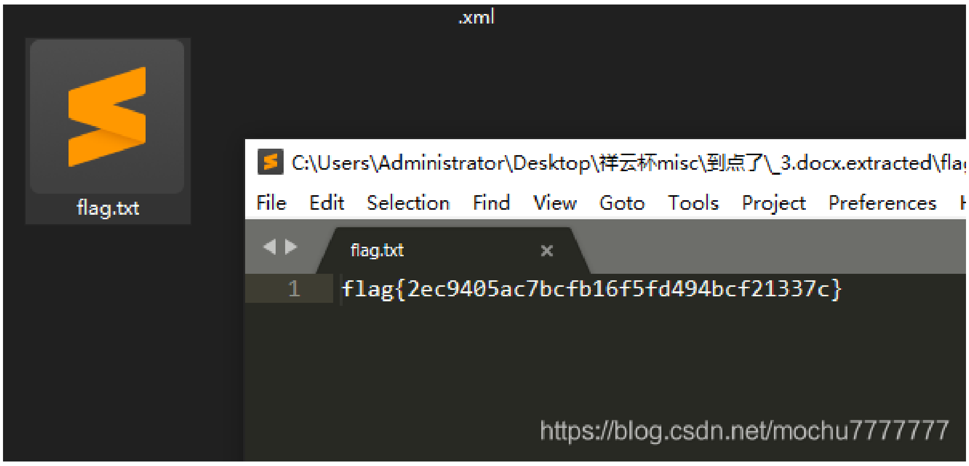
- Bitmapgrafik (*.BMP, *.RLE)
- Textdatei file (*.TXT, *.ASC) - Standardmethode
- Textdatei - kompatible Methode, HTML oder XML Datei
- Adobe Acrobat Datei (*.PDF)



Pa遷ort angeben

Bitte geben Sie das Pa遷ort an, das f黵
die Verschl黶selung der Daten verwendet
wurde.
Wenn keine Verschl黶selung verwendet
wurde, lassen Sie das Eingabefeld leer.

密码： `goodnightsweetie`



.xml
flag.txt
C:\Users\Administrator\Desktop\祥云杯misc\到点了_3.docx.extracted\fla
File Edit Selection Find View Goto Tools Project Preferences
flag.txt
https://blog.csdn.net/mochu7777777

```
flag{2ec9405ac7bcfb16f5fd494bcf21337c}
```

```
flag{2ec9405ac7bcfb16f5fd494bcf21337c}
```

xixixi

题目描述：

```
室友最近沉迷y神，又氪又肝，还ghs。为了他的身体着想，我把他的s图整没了。但我明明删了脚本啊，为什么还能被他发现......8说了，医院的空调真舒服~
```

`new.vhd`

VHD 是微软虚拟磁盘文件。

VHD（Microsoft Virtual Hard Disk format）。

目前可以使用Microsoft Virtual PC 2007 and Microsoft Virtual Server 2005以及Hyper-V对此格式进行操作，

VirtualBox也提供了对VHD的支持。

微软的VHD文件格式是一种虚拟机硬盘(virtual machine hard disk), 并可以被压缩成单个文件存放在宿主机器的文件系统上，主要包括虚拟机启动所需系统文件。

关于VHD的应用：Virtual PC是一种windows虚拟机，它可以虚拟各种版本的windows,一个windows应该装在一个硬盘分区上，而它是虚拟的windows，不可能单独划出一个硬盘分区给它安装，所以它启动所需系统文件都被压缩成一个VHD格式的文件放在硬盘上。

VHD格式还将用于Microsoft Windows Server 2008 R2和Microsoft Windows 7，包括hypervisor为基础的虚拟化技术- Hyper-V 。 Hyper-V 可以离线操作VHD — 使得管理员可以通过一个VHD文件，安全进入系统，管理员可以对虚拟文件（VHD)访问和执行一些离线的管理任务。

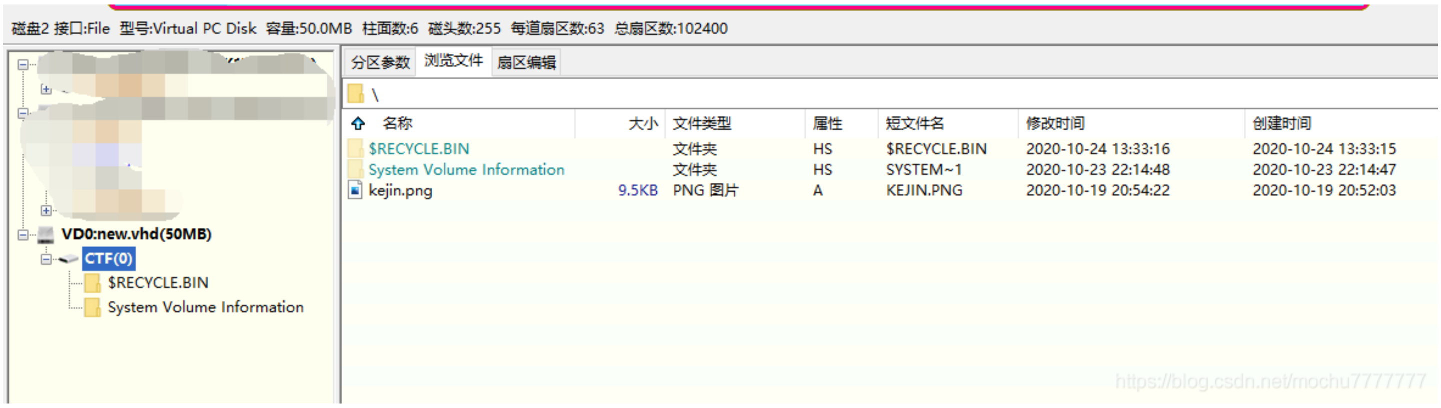
VHD 格式还应用在Windows Vista的Business, Enterprise and Ultimate 版本中，可以进行完整的系统备份。

WinMount 支持 挂载 VHD文件到虚拟盘，可以读取、修改、添加、删除虚拟盘的内容，并且支持保存修改到源始文件中。

WinMount支持将虚拟机硬盘镜像VHD(Virtual PC)、VDI(Virtual Box)、VMDK(VMWare)挂载为虚拟磁盘，并提供只读和可写两种打开方式。

可以使用`DiskGenius`或者Win7的磁盘管理进行挂载，建议使用`DiskGenius`挂载

`DiskGenius->磁盘->打开虚拟磁盘文件`



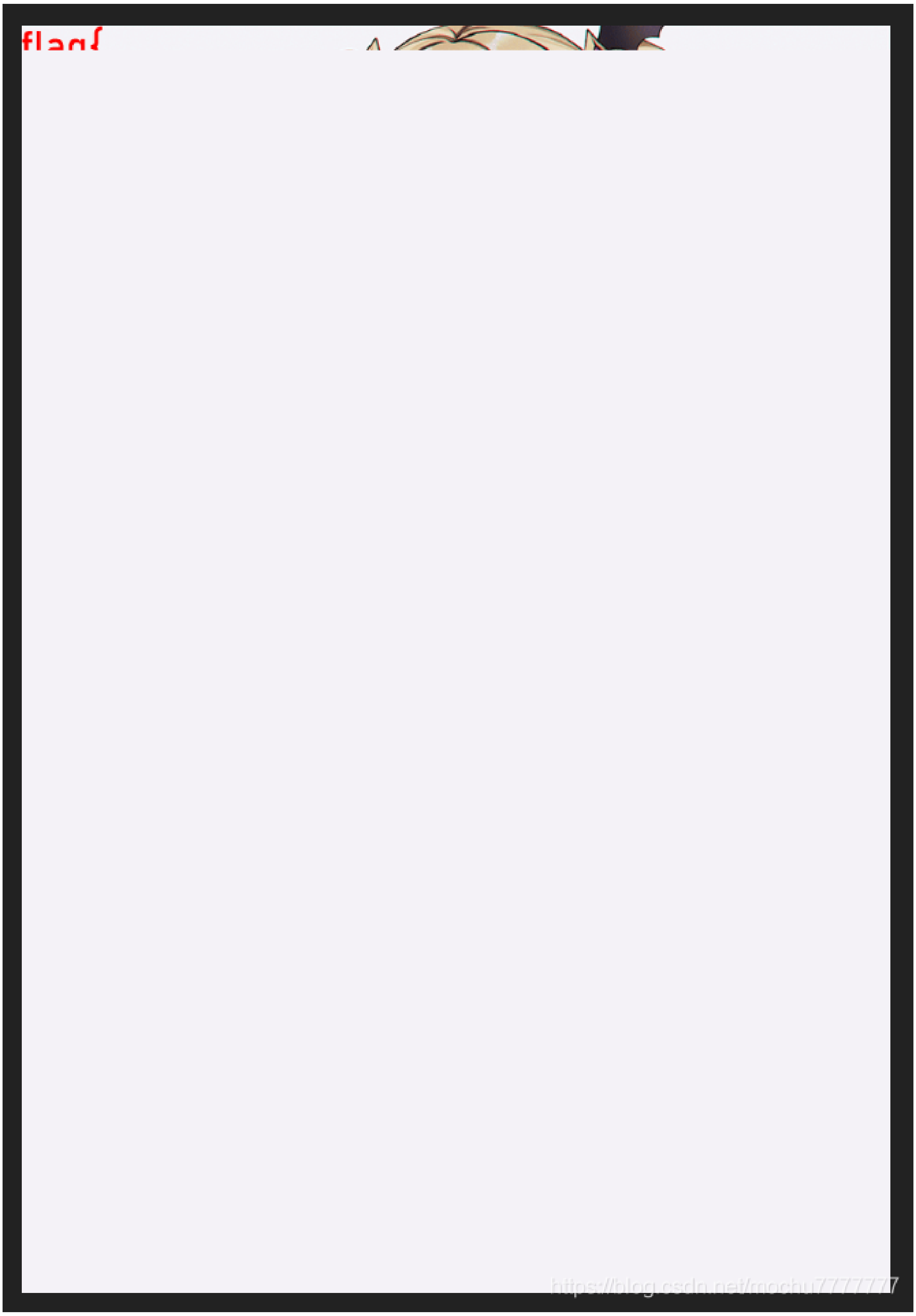
磁盘2 接口:File 型号:Virtual PC Disk 容量:50.0MB 柱面数:6 磁头数:255 每道扇区数:63 总扇区数:102400

[redacted]

VD0:new.vhd(50MB)
CTF(0)
$RECYCLE.BIN
System Volume Information

名称	大小	文件类型	属性	短文件名	修改时间	创建时间
$RECYCLE.BIN		文件夹	HS	$RECYCLE.BIN	2020-10-24 13:33:16	2020-10-24 13:33:15
System Volume Information		文件夹	HS	SYSTEM~1	2020-10-23 22:14:48	2020-10-23 22:14:47
kejin.png	9.5KB	PNG 图片	A	KEJIN.PNG	2020-10-19 20:54:22	2020-10-19 20:52:03

`kejin.png`



以及还有两个Py脚本

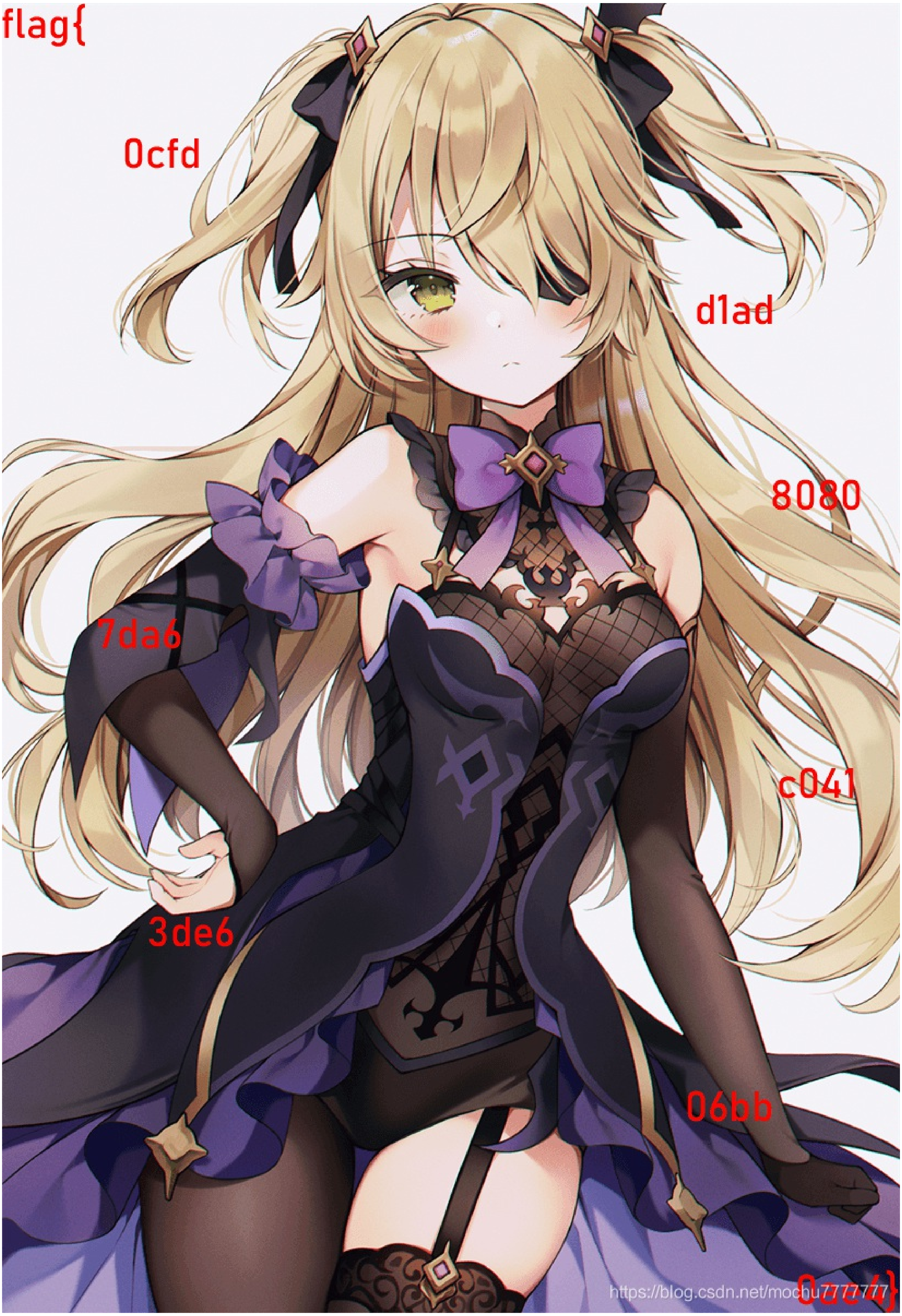
```
import struct

class FAT32Parser(object):
 def __init__(self, vhdFileName):
  with open(vhdFileName, 'rb') as f:
   self.diskData = f.read()
  self.DBR_off = self.GetDBRoff()
  self.newData = ''.join(self.diskData)

 def GetDBRoff(self):
  DPT_off = 0x1BE
  target = self.diskData[DPT_off+8:DPT_off+12]
  DBR_sector_off, = struct.unpack("<I", target)
  return DBR_sector_off * 512

 def GetFAT1off(self):
  target = self.diskData[self.DBR_off+0xE:self.DBR_off+0x10]
  FAT1_sector_off, = struct.unpack("<H", target)
  return self.DBR_off + FAT1_sector_off * 512

 def GetFATlength(self):
  target = self.diskData[self.DBR_off+0x24:self.DBR_off+0x28]
  FAT_sectors, = struct.unpack("<I", target)
  return FAT_sectors * 512

 def GetRootoff(self):
  FAT_length = self.GetFATlength()
  FAT2_off = self.GetFAT1off() + FAT_length
  return FAT2_off + FAT_length

 def Cluster2FAToff(self, cluster):
  FAT1_off = self.GetFAT1off()
  return FAT1_off + cluster * 4

 def Cluster2DataOff(self, cluster):
  rootDir_off = self.GetRootoff()
  return rootDir_off + (cluster - 2) * 512
```

```python
import struct
from xixi import FAT32Parser
from xixixi import Padding, picDepartList

def EncodePieces():
 global clusterList
 res = []
 Range = len(picDepartList)    # 58
 # GetRandomClusterList(n) - Generate a random cluster list with length n
 clusterList = GetRandomClusterList(Range)

 for i in range(Range):
  if i != Range - 1:
   newCRC = struct.pack("<I", clusterList[i+1])
   plainData = picDepartList[i][:-4] + newCRC
  else:
   plainData = picDepartList[i]

  # Show the first piece to him, hhh
  if i == 0:
   newPiece = plainData
  else:
   newPiece = ''
   key = clusterList[i] & 0xFE
   for j in plainData:
    newPiece += chr(ord(j) ^ key)
  # Padding() -- Fill to an integral multiple of 512 with \xFF
  res.append(Padding(newPiece))
 return res
```

参考上面给出的脚本进行还原，还原脚本参考的是`Timeline Sec`团队的脚本

原文地址：https://mp.weixin.qq.com/s/CP3-W8VcLokQNYMSbXw9wg

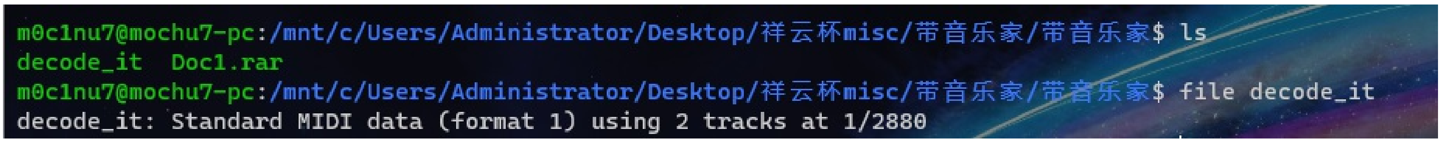
```python
# -*- coding: utf-8 -*-
# @Project: Hello Python!
# @File   : exp
# @Author : Tr0jAn <Tr0jAn@birkenwald.cn>
# @Date   : 2020-11-22
import struct
import binascii

class FAT32Parser(object):
  def __init__(self, vhdFileName):
    with open(vhdFileName, 'rb') as f:
      self.diskData = f.read()
    self.DBR_off = self.GetDBRoff()
    self.newData = ''.join(str(self.diskData))


  def GetDBRoff(self):
    DPT_off = 0x1BE
    target = self.diskData[DPT_off+8:DPT_off+12]
    DBR_sector_off, = struct.unpack("<I", target)
    return DBR_sector_off * 512


  def GetFAT1off(self):
    target = self.diskData[self.DBR_off+0xE:self.DBR_off+0x10]
    FAT1_sector_off, = struct.unpack("<H", target)
    FAT1_sector_off, = struct.unpack("<H", target)
    return self.DBR_off + FAT1_sector_off * 512


  def GetFATlength(self):
    target = self.diskData[self.DBR_off+0x24:self.DBR_off+0x28]
    FAT_sectors, = struct.unpack("<I", target)
    return FAT_sectors * 512


  def GetRootoff(self):
    FAT_length = self.GetFATlength()
    FAT2_off = self.GetFAT1off() + FAT_length
    return FAT2_off + FAT_length


  def Cluster2FAToff(self, cluster):
    FAT1_off = self.GetFAT1off()
    return FAT1_off + cluster * 4


  def Cluster2DataOff(self, cluster):
    rootDir_off = self.GetRootoff()
    return rootDir_off + (cluster - 2) * 512


def read(n):
    global key
    binary = b''
    for i in vhd.read(n):
        binary += (i ^ (key & 0xFE)).to_bytes(length=1, byteorder='big', signed=False)
    return binary


FAT = FAT32Parser("new.vhd")
vhd = open("new.vhd", "rb")
vhd.seek(0x27bae00)  # 定位磁盘中图片位置
flag = open("flag.png", "wb")
flag.write(vhd.read(8))  # 写入png头
key = 0
while True:
    d = read(8)
    length, cType = struct.unpack(">I4s", d)
    print(length, cType)  # length为数据长度，cType为数据块类型
    data = read(length)
    CRC = struct.unpack(">I", read(4))[0]
    print(CRC)
    rCRC = binascii.crc32(cType + data) & 0xffffffff
    print(rCRC)
    rDATA = struct.pack(">I", length) + cType + data + struct.pack(">I", rCRC)
    flag.write(rDATA)
    if CRC != rCRC:  # CRC错误的IDAT数据块
        b_endian = struct.pack(">I", CRC)
        clusterList = struct.unpack("<I", b_endian)[0]
        print(clusterList)
        vhd.seek(FAT.Cluster2DataOff(clusterList))
        key = clusterList & 0xFE
    if cType == b"IEND":
        break
```



```
flag{0cfdd1ad80807da6c0413de606bb0ae4}
```

带音乐家

`MIDI` 文件

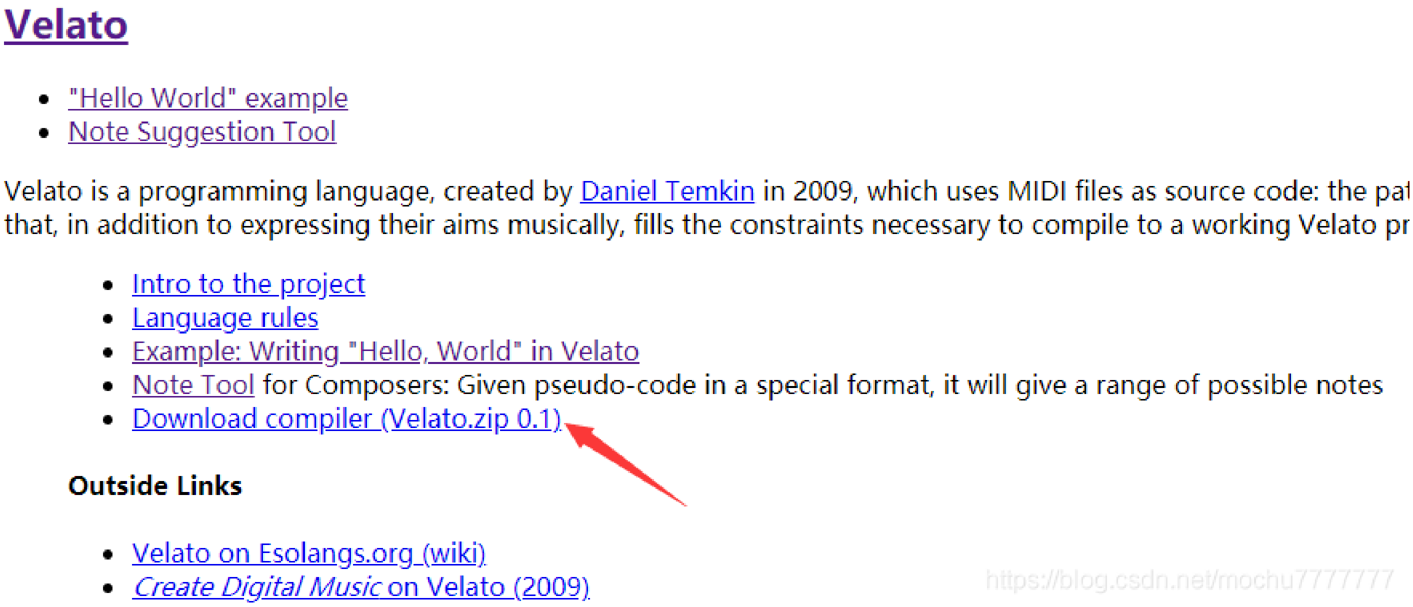


```
m0c1nu7@mochu7-pc:/mnt/c/Users/Administrator/Desktop/祥云杯misc/带音乐家/带音乐家$ ls
decode_it  Doc1.rar
m0c1nu7@mochu7-pc:/mnt/c/Users/Administrator/Desktop/祥云杯misc/带音乐家/带音乐家$ file decode_it
decode_it: Standard MIDI data (format 1) using 2 tracks at 1/2880
```

> `Velato语言` 使用 `MIDI` 文件作为源代码，音乐的模式决定程序命令

官网下载编译器

http://velato.net/



Velato

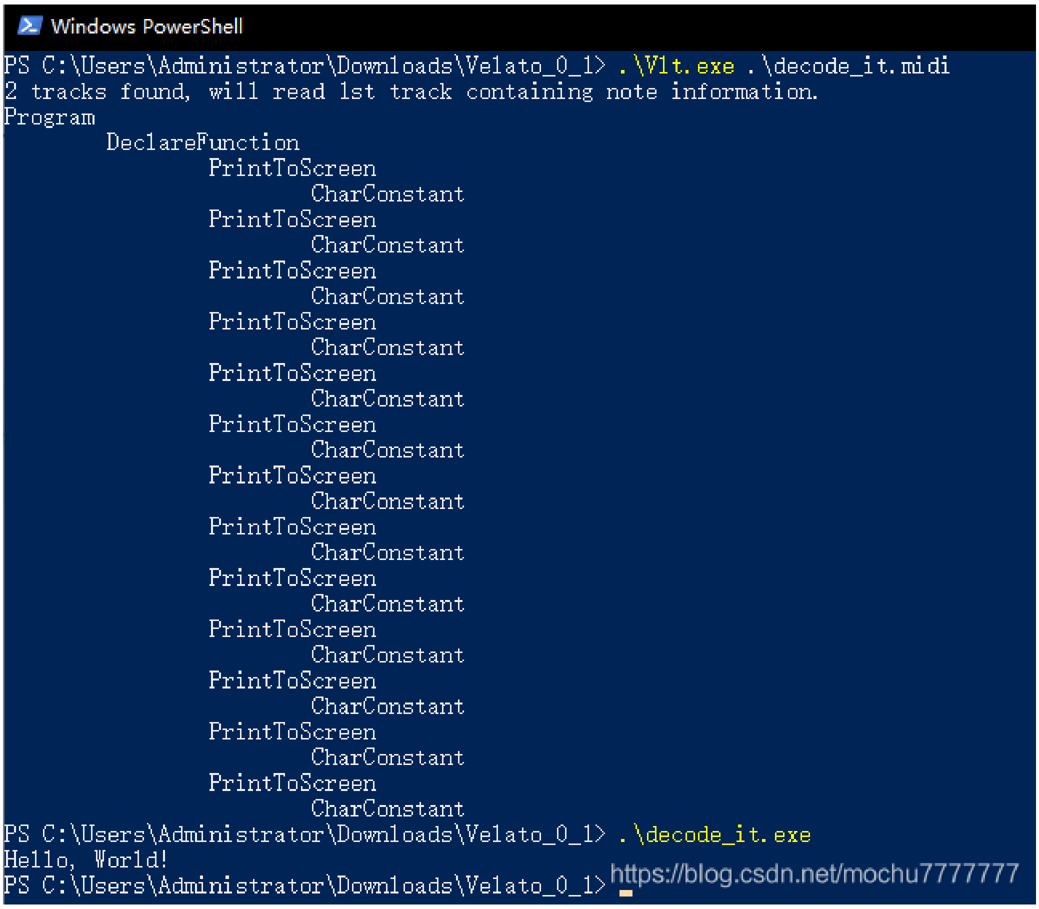
- "Hello World" example
- Note Suggestion Tool

Velato is a programming language, created by Daniel Temkin in 2009, which uses MIDI files as source code: the pat
that, in addition to expressing their aims musically, fills the constraints necessary to compile to a working Velato pr

- Intro to the project
- Language rules
- Example: Writing "Hello, World" in Velato
- Note Tool for Composers: Given pseudo-code in a special format, it will give a range of possible notes
- Download compiler (Velato.zip 0.1)

Outside Links

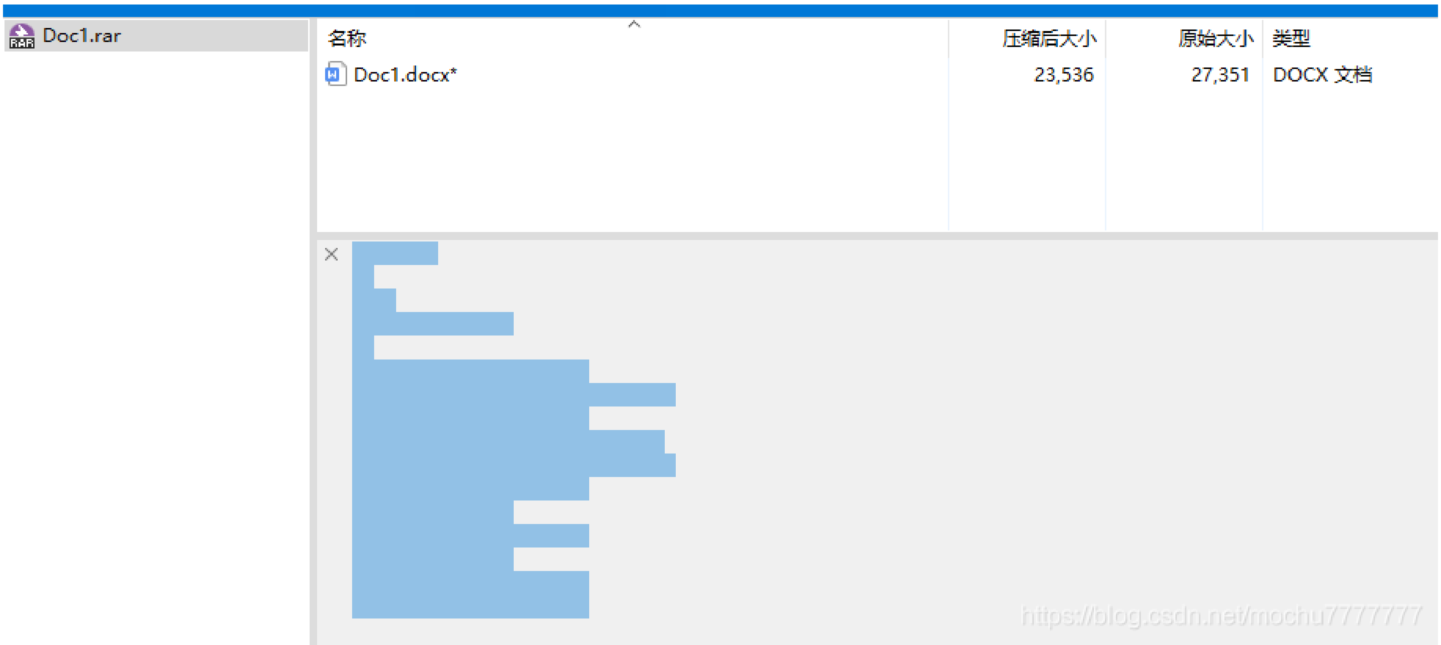
- Velato on Esolangs.org (wiki)
- *Create Digital Music* on Velato (2009)



```
PS C:\Users\Administrator\Downloads\Velato_0_1> .\Vlt.exe .\decode_it.midi
2 tracks found, will read 1st track containing note information.
Program
        DeclareFunction
                PrintToScreen
                        CharConstant
                PrintToScreen
                        CharConstant
                PrintToScreen
                        CharConstant
                PrintToScreen
                        CharConstant
                PrintToScreen
                        CharConstant
                PrintToScreen
                        CharConstant
                PrintToScreen
                        CharConstant
                PrintToScreen
                        CharConstant
                PrintToScreen
                        CharConstant
                PrintToScreen
                        CharConstant
                PrintToScreen
                        CharConstant
                PrintToScreen
                        CharConstant
                PrintToScreen
                        CharConstant
PS C:\Users\Administrator\Downloads\Velato_0_1> .\decode_it.exe
Hello, World!
PS C:\Users\Administrator\Downloads\Velato_0_1>
```

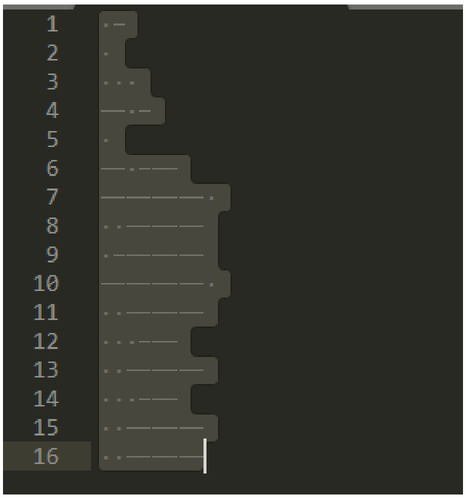
```
Hello, World!
```

`Doc1.rar`注释有东西



Doc1.rar

名称	压缩后大小	原始大小	类型
Doc1.docx*	23,536	27,351	DOCX 文档



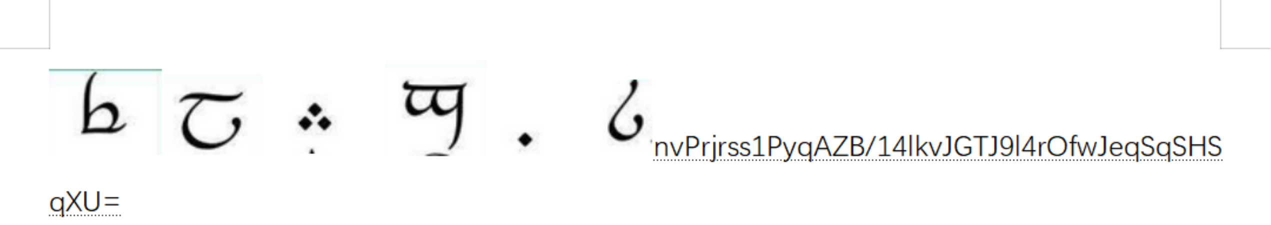
```
.-
.
...
-.-
.
-.--
----.
..---
.----
----.
..---
...--
..---
...--
..---
..---
```

摩斯，短的转为`.`，长的转为`-`

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

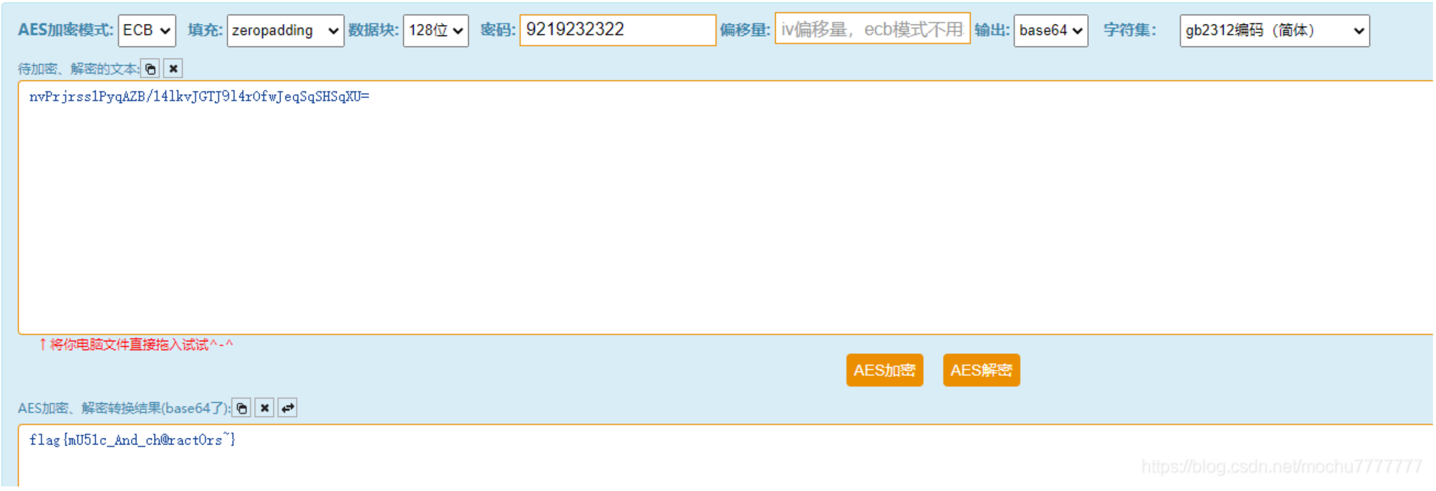
```
AESKEY9219232322
```

解压`Doc1.rar`，打开`Doc1.docx`(记得开启隐藏字符)



nvPrjrss1PyqAZB/14lkvJGTJ9l4rOfwJeqSqSHS
qXU=

```
nvPrjrss1PyqAZB/14lkvJGTJ9l4rOfwJeqSqSHSqXU=
```


AES加密模式: ECB 填充: zeropadding 数据块: 128位 密码: 9219232322 偏移量: iv偏移量，ecb模式不用 输出: base64 字符集: gb2312编码（简体）

待加密、解密的文本:

```
nvPrjrss1PyqAZB/14lkvJGTJ914r0fwJeqSqSHSqXU=
```

AES加密、解密转换结果(base64了):

```
flag{mU51c_And_ch@ract0rs~}
```

```
flag{mU51c_And_ch@ract0rs~}
```

Charles Sensor

等待大佬wp…orz