CTF 解题技能之 MISC 基础

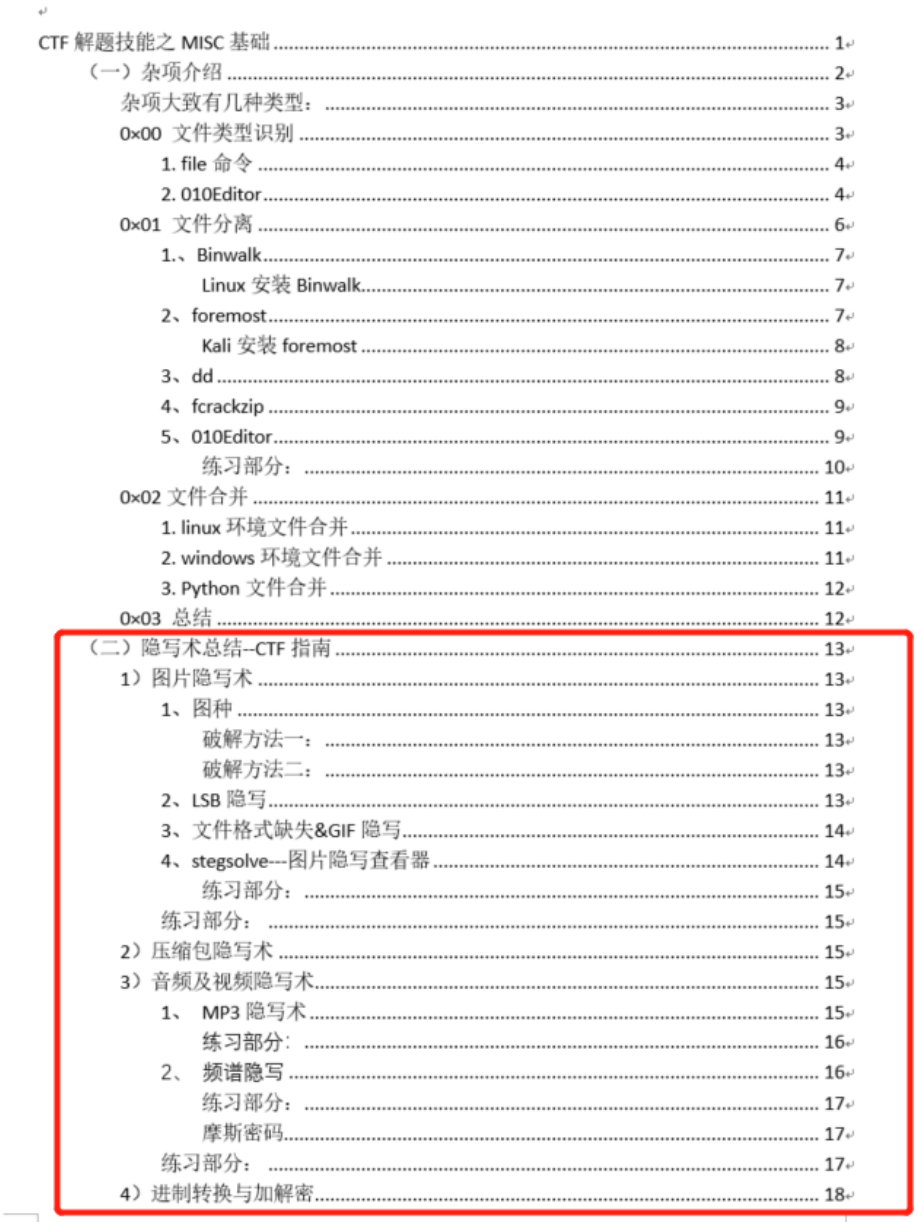
前言

以下是我近两个月对 MISC 的学习，学习思路大多来源于网络，因为学习过程查询的资料太多，部分资料记录了原文链接，部分资料原文链接忘了记录。

本人只是在前人学习的基础上做了系统的整理。

来源于网络，回馈于网络~

整理后资料分为两个部分《CTF 解题技能之 MISC 基础一》和《CTF 解题技能之 MISC 基础二》，目录结构如下：



继（一）杂项介绍，本文介绍红圈内的部分。

（二）隐写术总结--CTF 指南

1）图片隐写术

1、图种

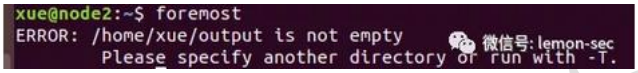
如何把一个压缩包隐藏在图片中：
例如一个 zip 压缩包（1.zip）+一个 jpg 图片（4fcefdaba56019d77b476e30a5558b47.jpg）结合为一张图片（output.jpg）
以下为 Windows 命令：

C:\Users\lenovo>copy/b C:\Users\lenovo\Desktop\1.zip +
C:\Users\lenovo\Desktop\4fcefdaba56019d77b476e30a5558b47.jpg output.jpg
Windows 输出：

破解方法一：

使用 kali Linux 的 binwalk 工具 检索图片文件里的其他文件

然后使用 Linux 的 foremost 工具 将图片中的 zip 文件分离出来



破解方法二：

直接将文件后缀改为.rar，然后解压，但对于隐写了多个文件可能会失败

2、LSB 隐写

LSB 隐写，也就是最低有效位（Least Significant Bit）。图片中的像数一般是由三原色组成，由这三种原色可以组成其他各种颜色，例如在 PNG 图片的储存中，每个颜色会有 8bit，LSB 隐写就是修改了像数中的最低的 1bit，写入加密信息，而人眼无法注意到前后的变化。

例如此图看起只是六只环保色猪头，但是其中包含了一张隐藏的二维码，我们可以通过工具 Stegsolve.jar 打开此图，然后通过下方的按钮切换到 Gray bits，可以看到左上角出现了隐写在该通道的二维码，扫描二维码即可得到 flag。

3、文件格式缺失&GIF 隐写

一张名为“此为 gif 图片.gif”的文件，打开发现了报错。

我们将其拖入 winhex 中查看。在 CTF 中有的时候会需要我们去修复图片，这对我们对于图片的文件结构要有了解。找到 gif 的文件格式，然后对照破损的文件对其进行修复。

我们可以用工具一帧一帧的观察图片。Stegsolve 就带有这种功能。

Stegsolve——Analyse——Frame Brower

4、stegsolve---图片隐写查看器

2 中的 LSB 隐写、3 中的一帧一帧的观察图片，都使用到了这个工具；

今天做 CTF 隐写术的题偶然发现一隐写图片查看的神器------stegsolve，分享给大家

stegsolve 下载地址：http://www.caesum.com/handbook/Stegsolve.jar

stegsolve 安装配置：配置好 Java 环境变量（就是需要安装 Java，然后配环境变量，具体的配置过程上网一搜一堆，这里就不赘述）

配置好环境之后直接打开就可以使用

stegsolve 功能简介：

上面是软件打开的界面，界面简单。主要供能为 analyse，下面对 Analyse 下面几个功能键作简单介绍：

File Format:文件格式，这个主要是查看图片的具体信息

Data Extract:数据抽取，图片中隐藏数据的抽取

Frame Browser:帧浏览器，主要是对 GIF 之类的动图进行分解，动图变成一张张图片，便于查看

Image Combiner:拼图，图片拼接。

练习部分：

杂项第十二题:闪的好快：stegsolve

练习部分：

杂项第一题: 签到题、

杂项第十一题: 多种方法解决

2）压缩包隐写术

看下面（六）压缩包解密的几种方法

3）音频及视频隐写术

音频相关的 CTF 题目主要使用的隐写策略，主要分为 MP3 隐写，LSB 隐写，波形隐写，频谱隐写等。

主要使用到的几个工具 ：

- Audacity，ocenaudio 音频编辑的工具软件
- Mp3stego 将需要加密的数据压缩加密隐藏在 MP3 文件中
- video to Picture 可以将视频的每帧提转换成图片
- QR Research ctf 中常用的二维码识别软件
- mageMagick 是一个免费的创建、编辑、合成图片的软件
- 以及一张摩斯密码对照表

摩尔斯电码表

字符	电码符号	字符	电码符号	字符	电码符号
A	•—	N	—•	1	•————
B	—•••	O	———	2	••———
C	—•—•	P	•——•	3	•••——
D	—••	Q	——•—	4	••••—
E	•	R	•—•	5	•••••
F	••—•	S	•••	6	—••••
G	——•	T	—	7	——•••
H	••••	U	••—	8	———••
I	••	V	•••—	9	————•
J	•———	W	•——	0	—————
K	—•—	X	—••—	?	••——••
L	•—••	Y	—•——	/	—••—•
M	——	Z	——••	(	—•——•

密码学具体内容：《密码学入门知识》

1、 MP3 隐写术

MP3 隐写两种方式：

第一种：题目中给了密码了，用 mp3stego 去解密

第二种：如果在题目中没有给 key，而附件只给了一个 MP3，那就有可是用 mp3stego 隐藏的数据，也有可能是在音轨的频谱中隐藏了数据。

MP3 隐写主要是使用 Mp3Stego 工具进行隐写，使用方法如下：

encode -E hidden_text.txt -P pass svega.wavsvega_stego.mp3

练习部分:

杂项第二十三题: 旋转跳跃

音频工具 MP3stego 使用（一）

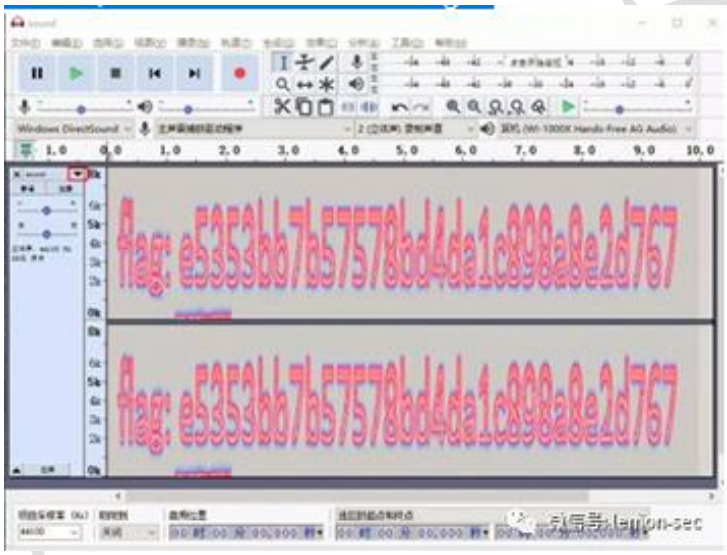
音频工具 MP3stego 使用（二）

2、 频谱隐写

频谱隐写：音频中的频谱隐写是将字符串隐藏在频谱中，此类音频通常会有一个较明显的特征，听起来是一段杂音或者比较刺耳。比如这题 Hear With Your Eyes。

我们使用 audacity 打开。

点开三角形（或 Shift+M)选择频谱图即可看到 Flag。



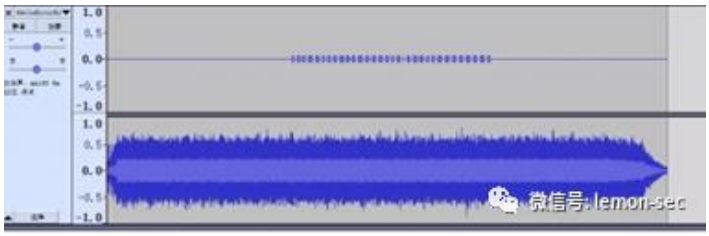
练习部分：

摩斯密码：杂项第二十八题：听首音乐

摩斯密码

这里需要我们熟悉音频分析工具的使用；

还有使用摩斯密码的比如这题 Hellokitty,打开文件后可以明显看到两段音轨不一样。



这段 MP3 正常播放一般是听不出来什么的，但是 Shift+M 选择分离立体声到单声道然后独奏播放第一栏的音轨，是不是超级熟悉。

我们将得到的摩斯码解码可以发现是串字符。

于是试试解 MD5，得到 flag 值。

练习部分：

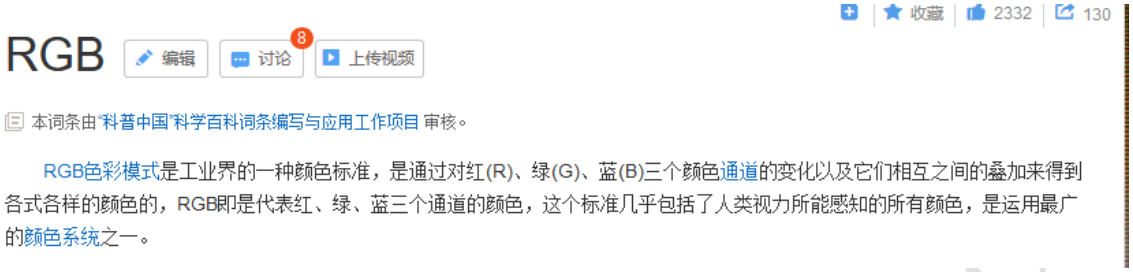
杂项第二十九题: ctf 练习---摩斯密码

4）进制转换与加解密

参考我另一文档《密码学入门知识》

5）RGB 值

RGB 值



在电脑中，RGB 的所谓“多少”就是指亮度，并使用整数来表示。通常情况下，RGB 各有 256 级亮度，用数字表示为从 0、1、2...直到 255。注意虽然数字最高是 255，但 0 也是数值之一，因此共 256 级。

常见颜色

颜色名称	红色值 Red	绿色值 Green	蓝色值 Blue
黑色	0	0	0
蓝色	0	0	255
绿色	0	255	0
青色	0	255	255
红色	255	0	0
亮紫色(洋红色)	255	0	255
黄色	255	255	0
白色	255	255	255

以上颜色为常用的基本颜色。

RGB 值：杂项第三十题: 好多数值（可参考练习，本地并未安装 yufu）

6）游戏隐写

通过游戏之类

练习部分：

杂项第十三题: come_game

7）颜文字

jjencode/aaencode（颜文字）

首先，

什么是 jjencode?

将 JS 代码转换成只有符号的字符串

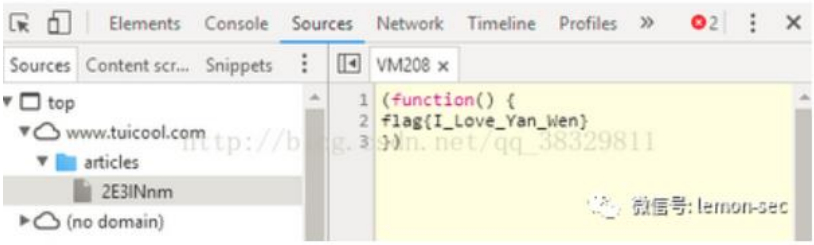
什么是 aaencode？

将 JS 代码转换成常用的网络表情

aaencode 加密：http://utf-8.jp/public/aaencode.html

解密方法：

可以直接利用浏览器的控制台输入密文，执行后即可解密。



原文链接：https://blog.csdn.net/qq_38329811/article/details/78186362

8）其他

1、PDF 隐写

Office 系列软件作为优秀的办公软件为我们提供了极大的便利，其中的 Word、Excel、PowerPoint 提供了许多在文档中隐藏数据的方法，比如批注、个人信息、水印、不可见内容、隐藏文字和定制的 XML 数据。今天我们涉及到的就是提到的隐藏文本功能。

利用 PDF 文件头添加额外信息，这个区域的信息会被 Adobe Acrobat Reader 阅读器忽略。

工具：wbStego4open

wbStego4open 会把插入数据中的每一个 ASCII 码转换为二进制形式，然后把每一个二进制数字再替换为十六进制的 20 或者 09，20 代表 0，09 代表 1。

最后，这些转换后的十六进制数据被嵌入到 PDF 文件中。查看用 wbStego4open 修改后的文件内容，会发现文件中已混入了很多由 20 和 09 组成的 8 位字节

2、DOC 隐藏

Doc 文件的本质是一个压缩文件，常见的隐藏文本的方式有两种，即：将字体隐藏或者设置同色字体，以下就是一个字体隐藏的例子

如上图所示，除了两行无用数据意外我们并不能看出有其他有价值的信息，判断 flag 可能是被隐藏，需要开启文本显示功能，查看是否是我们猜测的那样。点击左上角　　文件-选项，打开 Word 选项对话框，在“显示”中勾选隐藏文字选项

3、数据包隐写术

数据包隐写术，就是将所要传达的信息和文件，以流量包的形式下发给参赛选手，参赛选手要从流量包中自行提取出所需要的文件或者相关内容进行解题。比较常用的工具是 wireshark。关于此类部分的详细介绍，大家可以访问这个网址：https://ctf-wiki.github.io/ctf-wiki/misc/traffic/data/

数据包隐写术目前两种考察行为：

①、flag 或者关键信息直接隐藏在流量包中

②、flag 相关文件隐藏在流量包中，需要分离文件

原有链接：https://www.cnblogs.com/xww115/p/11242631.html

练习部分：

数据包隐写　杂项第四题: telnet、

4、linux 隐写

Grep 查看文件或搜索 flag、key

练习部分：

杂项第十五题: linux

5、其他练习部分：

杂项第八题: 猜

杂项第九题: 宽带信息泄露

（三）PNG 文件格式详解

PNG 是 20 世纪 90 年代中期开始开发的图像文件存储格式，其目的是替代 GIF 和 TIFF 文件格式，同时增加一些 GIF 文件格式所不具备的特性。流式网络图形格式(Portable Network Graphic Format，PNG)名称来源于非官方的“PNG’s Not GIF”，是一种位图文件(bitmap file)存储格式，读成“ping”。PNG 用来存储灰度图像时，灰度图像的深度可多到 16 位，存储彩色图像时，彩色图像的深度可多到 48 位，并且还可存储多到 16 位的 α 通道数据。PNG 使用从 LZ77 派生的无损数据压缩算法。（说白了这就是一种方便的、适于网络传播的轻便图片文件格式）

文件结构

PNG 图像格式文件由文件署名和数据块(chunk)组成。

PNG 文件格式规范制定的 10 个辅助数据块是：

1. 背景颜色数据块 bKGD(background color)。
2. 基色和白色度数据块 cHRM(primary chromaticities and white point)。所谓白色度是指当 R＝G＝B＝最大值时在显示器上产生的白色度。
3. 图像 γ 数据块 gAMA(image gamma)。
4. 图像直方图数据块 hIST(image histogram)。
5. 物理像素尺寸数据块 pHYs(physical pixel dimensions)。
6. 样本有效位数据块 sBIT(significant bits)。
7. 文本信息数据块 tEXt(textual data)。
8. 图像最后修改时间数据块 tIME (image last-modification time)。
9. 图像透明数据块 tRNS (transparency)。
10. 压缩文本数据块 zTXt (compressed textual data)。

关键数据块、辅助数据块和专用公共数据块(special-purpose public chunks)综合下表中：

数据块符号	数据块名称	多数据块	可选否	位置限制
IHDR	文件头数据块	否	否	第一块
cHRM	基色和白色点数据块	否	是	在PLTE和IDAT之前
gAMA	图像γ数据块	否	是	在PLTE和IDAT之前
sBIT	样本有效位数据块	否	是	在PLTE和IDAT之前
PLTE	调色板数据块	否	是	在IDAT之前
bKGD	背景颜色数据块	否	是	在 PLTE 之后 IDAT 之前
hIST	图像直方图数据块	否	是	在 PLTE 之后 IDAT 之前
tRNS	图像透明数据块	否	是	在 PLTE 之后 IDAT 之前
oFFs	(专用公共数据块)	否	是	在IDAT之前
pHYs	物理像素尺寸数据块	否	是	在IDAT之前
sCAL	(专用公共数据块)	否	是	在IDAT之前
IDAT	图像数据块	是	否	与其他IDAT连续
tIME	图像最后修改时间数据块	否	是	无限制
tEXt	文本信息数据块	是	是	无限制
zTXt	压缩文本数据块	是	是	无限制
fRAc	(专用公共数据块)	是	是	无限制
gIFg	(专用公共数据块)	是	是	无限制
gIFt	(专用公共数据块)	是	是	无限制
gIFx	(专用公共数据块)	是	是	无限制
IEND	图像结束数据	否	否	最后一个数据块

一个例子

为了便于研究，我在本地找了个 24x24 像素的图片：
用十六进制打开后是这样的：

```
1  0000000: 8950 4e47 0d0a 1a0a 0000 000d 4948 4452  .PNG........IHDR
2  0000010: 0000 0018 0000 0018 0806 0000 00e0 773d  ..............w=
3  0000020: f800 0000 1974 4558 7453 6f66 7477 6172  .....tEXtSoftwar
4  0000030: 6500 4164 6f62 6520 496d 6167 6552 6561  e.Adobe ImageRea
5  0000040: 6479 71c9 653c 0000 0344 4944 4154 78da  dyq.e<...DIDATx.
6  0000050: b454 4b48 5b51 10bd 792f 2646 a346 7411  .TKH[Q..y/&F.Ft.
7  0000060: 450b cac3 6840 8c14 0242 8542 e9aa ab42  E...h@...B.B...B
8  0000070: 5785 8614 c428 b4d0 5569 b108 5dbb 29b4  W....(..Ui..].).
9  0000080: 1b05 a5ab 6cb3 2a14 0ab5 8b42 75a3 d188  ....l.*....Bu...
10 0000090: 82c6 0ff1 4320 a2e2 2f7e 12ed 9c47 e671  ....C ../~...G.q
11 00000a0: 8d2f 8950 3a30 dcc7 bb33 67e6 cee7 5826  ./.P:0...3g...X&
12 00000b0: 2626 8499 288a a2ab d56a d555 55d5 57d7  &&..(....j.UU.W.
13 00000c0: d7d7 be6c 36fb fef2 f232 45a7 b8ba ba12  ...l6....2E.....
14 00000d0: c160 5014 13ab d94f 8bc5 72e3 24e0 1e9f  .`P....O..r.$...
15 00000e0: cff7 b9ae ae4e 4c4e 4eda 3299 4c00 777c  .....NLNN.2.L.w|
16 00000f0: 5f4c 1472 16f9 9a07 2e6a 6b6b 875a 5b5b  _L.r.....jkk.Z[[
17 0000100: 454d 4d8d e8ea ea7a 4eff 3ce2 8ea2 3018  EMM....zN.<...0.
18 0000110: 94cb c28a 7f04 765f d3b4 27ec d0d8 d8a8  ......v_..'.....
19 0000120: 5655 55bd 639b 9201 a8b6 a8af 516b 9bcd  VUU.c.......Qk..
20 0000130: 26ca caca f46f 0020 7bb7 db6d 38d8 ed76  &....o. {..m8..v
21 0000140: d1d1 d1f1 82ee 34d8 940c 0023 00c2 11e0  ......4....#....
22 0000150: 1c20 975d 2781 3d75 381c 379c 9a9b 9b55  . .]'.=u8.7....U
23 0000160: 2ad7 1012 bb73 0028 c073 1373 8f02 3c68  *....s.(.s.s..<h
24 0000170: 6b6b fb44 25b9 e554 5e5e 2eba bbbb f18a  kk.D%..T^^......
25 0000180: c791 4844 2b16 c012 0e87 fd04 faac bebe  ..HD+...........
```

```
26 0000190: be85 cad1 e272 b97c 04a0 e245 4ea7 530f  .....r.|...EN.S.
27 00001a0: 5c48 8e8f 8f45 3a9d d6f5 f0f0 7091 747b  \H...E:.....p.t{
28 00001b0: 7777 779d feff a649 0b07 0281 acb5 baba  www....I........
29 00001c0: fa97 dfef 7710 7041 309e 2c79 ba20 4800  ....w.pA0.,y. H.
30 00001d0: 9a13 2f81 7a11 7479 79b9 3f16 8ba1 ae63  ../.z.tyy.?....c
31 00001e0: caf9 f979 120b 83c5 81e6 03e3 8e4f 59e5  ...y.........OY.
32 00001f0: a010 5e3c f405 2705 dbd6 7b90 4aa5 1ecd  ..^<..'...{.J...
33 0000200: cdcd 250e 0e0e c4c5 c585 a080 fa49 0686  ..%..........I..
34 0000210: 1303 ca81 7007 1bf6 c1f7 d9d9 9958 5b5b  ....p........X[[
35 0000220: 1384 f73a 140a 7de3 3d58 27e9 8d46 a371  ...:..}.=X'..F.q
36 0000230: 0491 c14a 89fc 1204 01f8 d4d4 1461 87be  ...J.........a..
37 0000240: e46f 7262 6363 a377 6666 6671 6f6f cfc8  .orbcc.wfffqoo..
38 0000250: dc6c cb65 651b 3439 1e8f 03bc 8fc0 c70a  .l.ee.49........
39 0000260: 5145 3291 483c 5c58 5888 a251 a5c0 5989  QE2.H<\XX..Q..Y.
40 0000270: f8c4 d6d6 5676 7a7a fae5 c0c0 c0f8 ad3d  ....Vvzz.......=
41 0000280: c823 ac14 3df3 c3d1 d191 d1e4 620a 5f94  .#..=.......b._.
42 0000290: 6673 7333 3238 38f8 b520 9be6 3887 ff79  fss3288.. ..8..y
43 00002a0: 989b f85f 7e3f e4a4 b0f1 74af 1525 3b36  ..._~?....t..%;6
44 00002b0: 8423 ed45 6745 4585 012c 4f92 3c41 fc02  .#.EgEE..,O.<A..
45 00002c0: d008 ed90 7774 74d4 5694 4df9 248e f182  ....wtt.V.M.$...
46 00002d0: 3278 4ccd 9aca 8161 039f caca 4a80 9b52  2xL....a....J..R
47 00002e0: b855 1e49 3a55 32f6 c209 60bc 4068 3aa6  .U.I:U2...`.@h:.
48 00002f0: 8b1a 9aa6 041c 4429 3a6f 7150 3000 9d9d  ......D):oqP0...
49 0000300: 641e 33ed 8114 a485 0238 4178 703c 3939  d.3......8Axp<99
50 0000310: 11c9 6452 acac acfc 248e 798b 9d21 b0e1  ..dR....$.y..!..
```

```
51 0000320: f6f6 f6fe a6a6 2683 5ec0 5b94 948f eec3  ......&.^.[.....
52 0000330: 055f c001 3015 5838 64bc baba 1add d9d9  ._..0.X8d.......
53 0000340: 01f0 0f76 a0cd 7d33 3f3f 3f42 73ff d1e3  ...v..}3???Bs...
54 0000350: f104 1b1a 1a54 f890 6805 a748 9a92 3fb4  .....T..h..H..?.
55 0000360: e6df 691f dcfb fbfb 2366 19e5 64fb f4f4  ..i.....#f..d...
56 0000370: b46f 7676 7664 6969 6998 5ed1 43fd 1837  .ovvvdiii.^.C..7
57 0000380: a5eb bb50 c2bf 8822 feb3 fc15 6000 74fe  ...P..."....`.t.
58 0000390: 7622 c159 82da 0000 0000 4945 4e44 ae42  v".Y......IEND.B
59 00003a0: 6082 0a                                  `..
```

接下来我们试着分析一下：

首先是八个字节的文件头标志，标识着 png 文件：

1	8950 4e47 0d0a 1a0a

接下来的地方就是 IHDR 数据块了：

找到 IHDR，在这之后的八个 bit 就是宽高的值：这个在逆向中经常会遇到。

下面是 IHDR 数据块的实际内容

0000 0018 图像的宽，24 像素

0000 0018 图像的高，24 像素

练习部分：

杂项第三题: 隐写

杂项第十六题: 隐写 3

（四）根据文件头数据判断文件类型

现有一文件，其扩展名未知或标记错误。假设它是一个正常的、非空的文件，且将扩展名更正后可以正常使用，那么，如何判断它是哪种类型的文件？

在后缀未知，或者后缀被修改的文件，依然通过文件头来判断该文件究竟是什么文件类型。我们可以使用一个文本编辑工具如 UltraEdit 打开文件（16 进制模式下），然后看文件头是什么字符，以下是常见文件类型的文件头字符(16 进制)，希望对你有帮助：

JPEG (jpg)，文件头：FFD8FF

PNG (png)，文件头：89504E47

GIF (gif)，文件头：47494638

TIFF (tif)，文件头：49492A00

Windows Bitmap (bmp)，文件头：424D

CAD (dwg)，文件头：41433130

Adobe Photoshop (psd)，文件头：38425053

Rich Text Format (rtf)，文件头：7B5C727466

XML (xml)，文件头：3C3F786D6C

HTML (html)，文件头：68746D6C3E

Email [thorough only] (eml)，文件头：44656C69766572792D646174653A

Outlook Express (dbx)，文件头：CFAD12FEC5FD746F

Outlook (pst)，文件头：2142444E

MS Word/Excel (xls.or.doc)，文件头：D0CF11E0

MS Access (mdb)，文件头：5374616E644617264204A

WordPerfect (wpd)，文件头：FF575043

Postscript (eps.or.ps)，文件头：252150532D41646F6265

Adobe Acrobat (pdf)，文件头：255044462D312E

Quicken (qdf)，文件头：AC9EBD8F

Windows Password (pwl)，文件头：E3828596

ZIP Archive (zip)，文件头：504B0304

RAR Archive (rar)，文件头：52617221

Wave (wav)，文件头：57415645

AVI (avi)，文件头：41564920

Real Audio (ram)，文件头：2E7261FD

Real Media (rm)，文件头：2E524D46

MPEG (mpg)，文件头：000001BA

MPEG (mpg)，文件头：000001B3

Quicktime (mov)，文件头：6D6F6F76

Windows Media (asf)，文件头：3026B2758E66CF11

MIDI (mid)，文件头：4D546864

练习部分：

杂项第五题：眼见非实(ISCCCTF)

（五）压缩包解密的几种方法

压缩包解密通用方法：

1.用 winhex 打开，搜索字符 pass 、 key 等，查看是否有含有压缩包密码

2.如果要爆破：先 0-6 位数字来一遍

3.如果爆破不成功可以根据题意或者社工 猜密码组合。例如某用户名叫王方，密码就有可能是 wangfang123.

4.伪加密

5.明文攻击

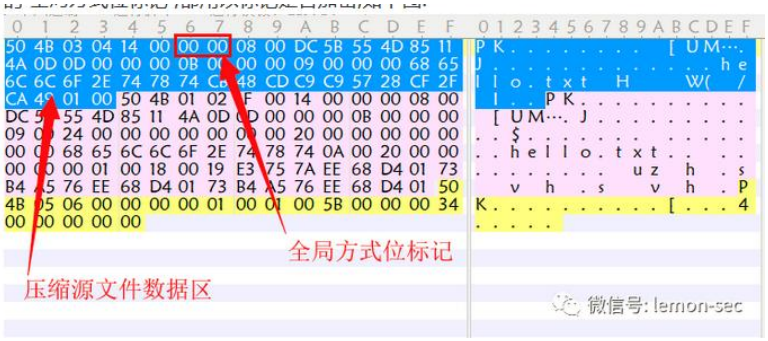
6.crc32 爆破

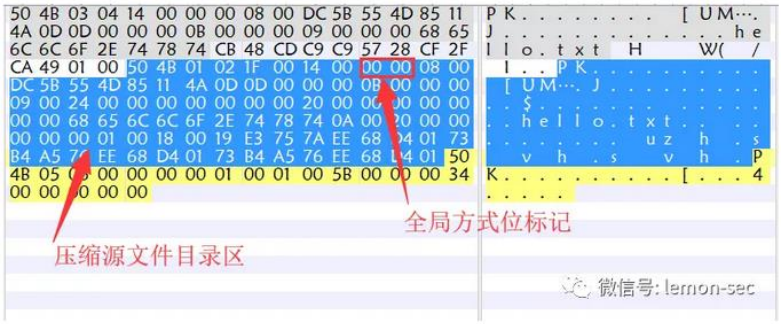
1、zip 伪加密

zip 文件是由 3 部分组成

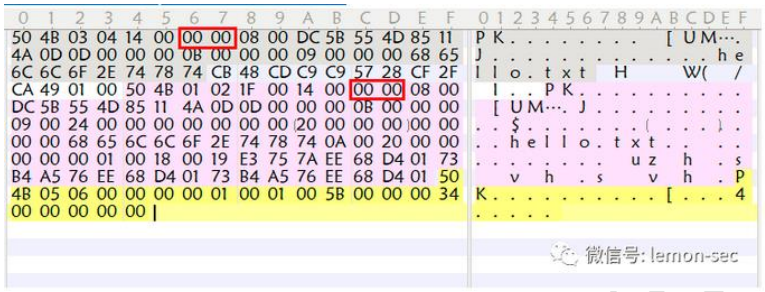
压缩源文件数据区+压缩源文件目录区+压缩源文件目录结束标志

在压缩源文件数据区有个 2 字节的 全局方式位标记 ，在压缩源文件目录区也有个 2 字节的 全局方式位标记 ，都用以标记是否加密，如下图：

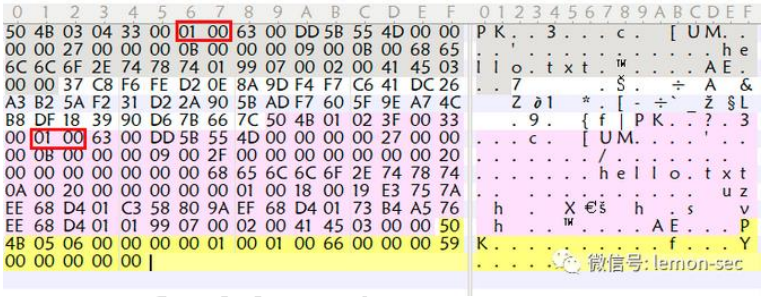




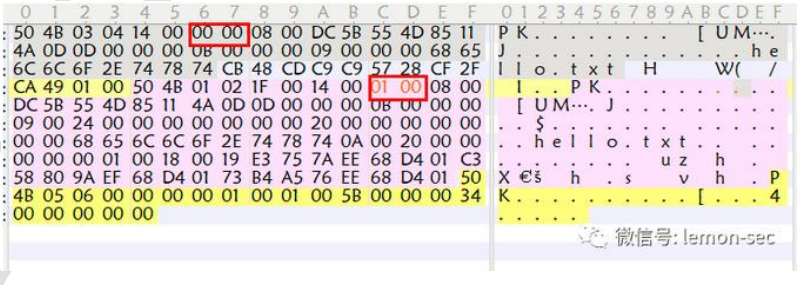
若是没有加密的 zip,两处标记都是 00 00



加密的 zip 两次都不是 00 00,好像不同版本的压缩软件或是算法,这里的值会不同,我看之前网上多见的是 09 00



若把未加密的 zip 压缩源文件目录区的全局方式位标记改为 01 00 (或者 09 00),就会被压缩软件认为是已加密,即所谓的伪加密



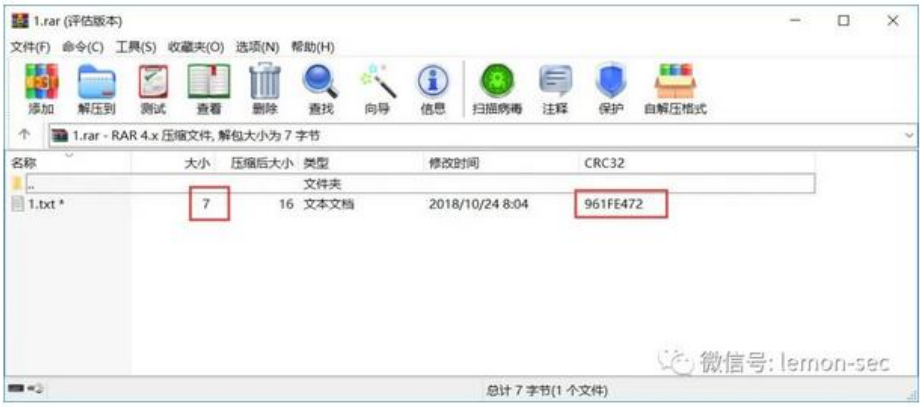
破解伪加密的 zip,只要把压缩源文件目录区的全局方式位标记改为 00 00 即可解压。

2、crc32 碰撞：

CRC 校验实用程序库 在数据存储和数据通讯领域，为了保证数据的正确，就不得不采用检错的手段。在诸多检错手段中，CRC 是最著名的一种。CRC 的全称是循环冗余校验。

总之每个文件都有唯一的 CRC32 值，即便数据中一个 bit 发生变化，也会导致 CRC32 值不同。若是知道一段数据的长度和 CRC32 值，便可穷举数据，与其 CRC32 对照，以此达到暴力猜解的目的。但通常只适用于较小文本文件。

比如这里有一个加密的 rar，直接双击就可以看见其中信息，而且我知道其中全是数字，便可写脚本爆破

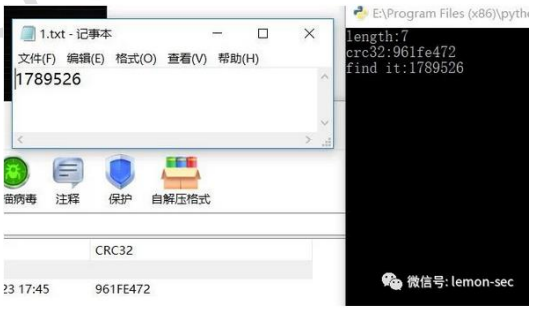


可以用 python 脚本爆破，也可以使用工具：AccentRPR

```
from zlib import crc32
 import random
char='0123456789'
def crc32_f(data):
    return hex(crc32(data)&0xffffffff)[2:10]
length=input('length:')
crc32_=raw_input('crc32:').lower()
while True:
    text=''
    for i in range(length):
        text+=char[random.randint(0,len(char)-1)]
    if crc32_f(text)==crc32_:
        raw_input('find it:'+text)
        exit
```

python2 中的 crc32()计算的 crc32 值是有符号的，所以&0xffffffff 转换成无符号数值

由于全是数字，7 位也不算太高，跑了 3 分钟左右便出结果了



已知明文攻击即 crc32 爆破：

所谓明文攻击就是已经通过其他手段知道 zip 加密文件中的某些内容，比如在某些网站上发现它的 readme.txt 文件，或者其他文件，这时就可以尝试破解了，例如我今天刚做了一个明文攻击的小题目：

源文件是：明文攻击.zip,但是一看发现 zip 文件里面有一个之前看到过的 hash.exe，和一些其他加密文件，所以就将已经拥有的 hash.exe 压缩成 zip，注意打包完成后，需要确认二者采用的压缩算法相同。一个简单的判断方法是用 winRAR 打开文件，同一个文件压缩后的体积是否相同。如果基本相同，并且 crc32 也必须相同，可以说明你用的压缩算法是正确

的。如果不同，就尝试另一种压缩算法。

接下来就用神器 archpr 进行明文破解，注意选择的明文是 hash.zip(特别注意 archpr5.3 版本找到密钥之后有提示而 5.4 版本没有提示)

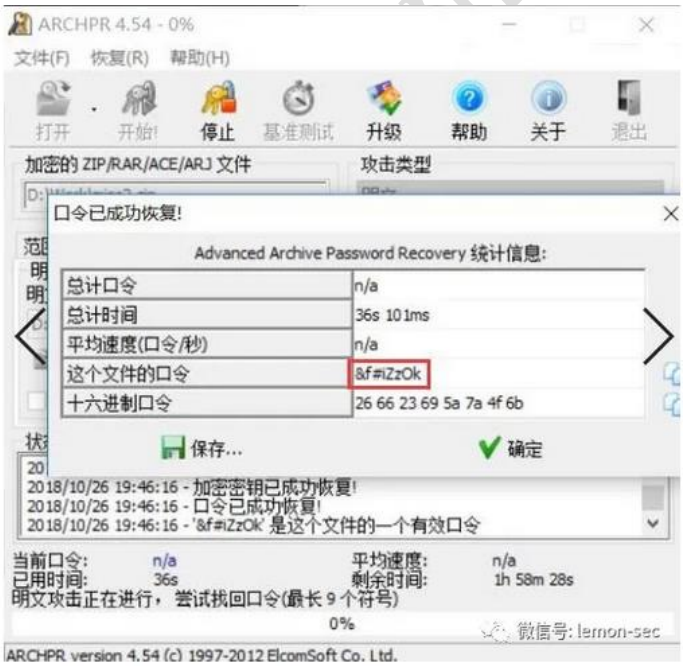
ARCHPR4.5.4 找回口令不行。ARCHPR 较 4.5.3 快。

如果得到了加密压缩包中的某个文件，那么就可以通过明文攻击来获取压缩密码

这里得到了 163264.txt 文件，以及一个加密的 misc2.zip，misc2.zip 中的文件即是我们得到的这个文件，先将已知的文件压缩，从 CRC32 也可以看出两个文件是一样的

然后在软件中选择明文（plain-text），填入路径点击开始

我跑了大约一两分钟



如果出现以下错误，那么一般就是你使用的压缩软件或是压缩算法和待破解的压缩包所使用的软件不同，可以用 WinRAR，7z，2345 好压等都试一下

3、直接爆破：

AccentRPR 利用了 GPU 爆破，速度还是比较快的。

1.暴力:选择密码范围,长度等,由软件组合生成密码进行爆破

2.掩码:知道密码中的一部分,只需按规则构造其余部分

3.字典:通常是多数用户常用的一些密码集合,导入字典文件用其中的密码进行爆破

推荐一款 windows 下的软件

练习部分：

杂项第三十一题: 神秘的文件