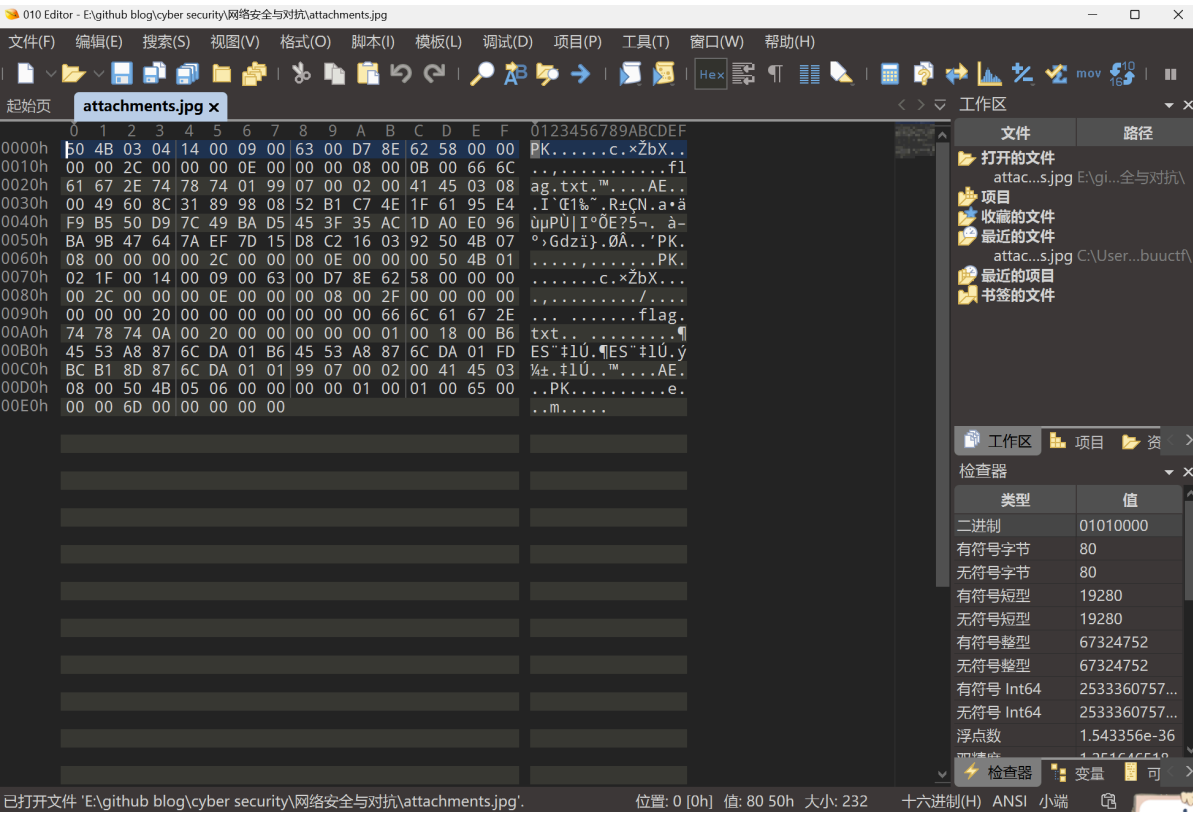


# 网络安全与对抗 第二讲 MISC Flag is here

首先我们将题目给的附件解压出来，jpg后缀可知这是一张图片

 attachments.jpg	2024/9/27 16:04	JPG 文件	1 KB
---	-----------------	--------	------

常规思路，我们将图片丢进010 editor看看



一眼看到50 4B 03 04立刻反应过来这是一个zip文件的头文件标记

## Zip文件格式（16进制）

```

50 4B 03 04 14 00 00 00 08 00 20 9E 66 4F F2 1B
0F 4A 0E 00 00 00 0C 00 00 00 08 00 00 00 66 6C
61 67 2E 74 78 74 4B CB 49 4C AF 36 34 32 36 31
35 AB 05 00 50 4B 01 02 1F 00 14 00 00 00 08 00
20 9E 66 4F F2 1B 0F 4A 0E 00 00 00 0C 00 00 00
08 00 24 00 00 00 00 00 00 00 20 00 00 00 00 00
00 00 66 6C 61 67 2E 74 78 74 0A 00 20 00 00 00
00 00 01 00 18 00 44 D0 15 2E 98 94 D5 01 44 D0
15 2E 98 94 D5 01 A4 1E 91 25 98 94 D5 01 50 4B
05 06 00 00 00 00 01 00 01 00 5A 00 00 00 34 00
00 00 00 00


```

[https://blog.csdn.net/qq\\_45861039](https://blog.csdn.net/qq_45861039)

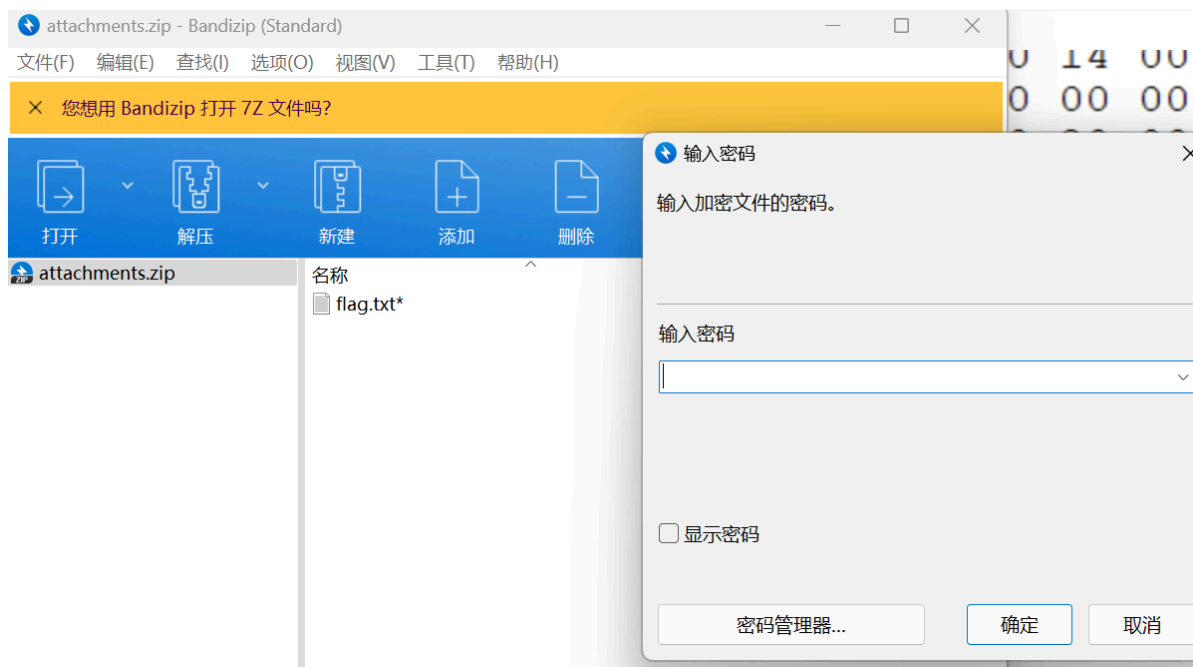
这就是一个zip文件的格式。

- 1 压缩源文件数据区：
- 2 50 4B 03 04：这是头文件标记（0x04034b50）
- 3 14 00：解压文件所需 pkware 版本
- 4 00 00：全局方式位标记（有无加密）
- 5 08 00：压缩方式
- 6 20 9E：最后修改文件时间
- 7 66 4F：最后修改文件日期
- 8 F2 1B 0F 4A：CRC-32校验（4A0F1BF2）
- 9 0E 00 00 00：压缩后尺寸
- 10 0C 00 00 00：未压缩尺寸
- 11 08 00：文件名长度
- 12 00 00：扩展记录长度
- 13 66 6C 61 67 2E 74 78 74：文件名（不定长）
- 14 4B CB 49 4C AF 36 34 32 36 31 35 AB 05 00：文件flag.txt压缩后的数据

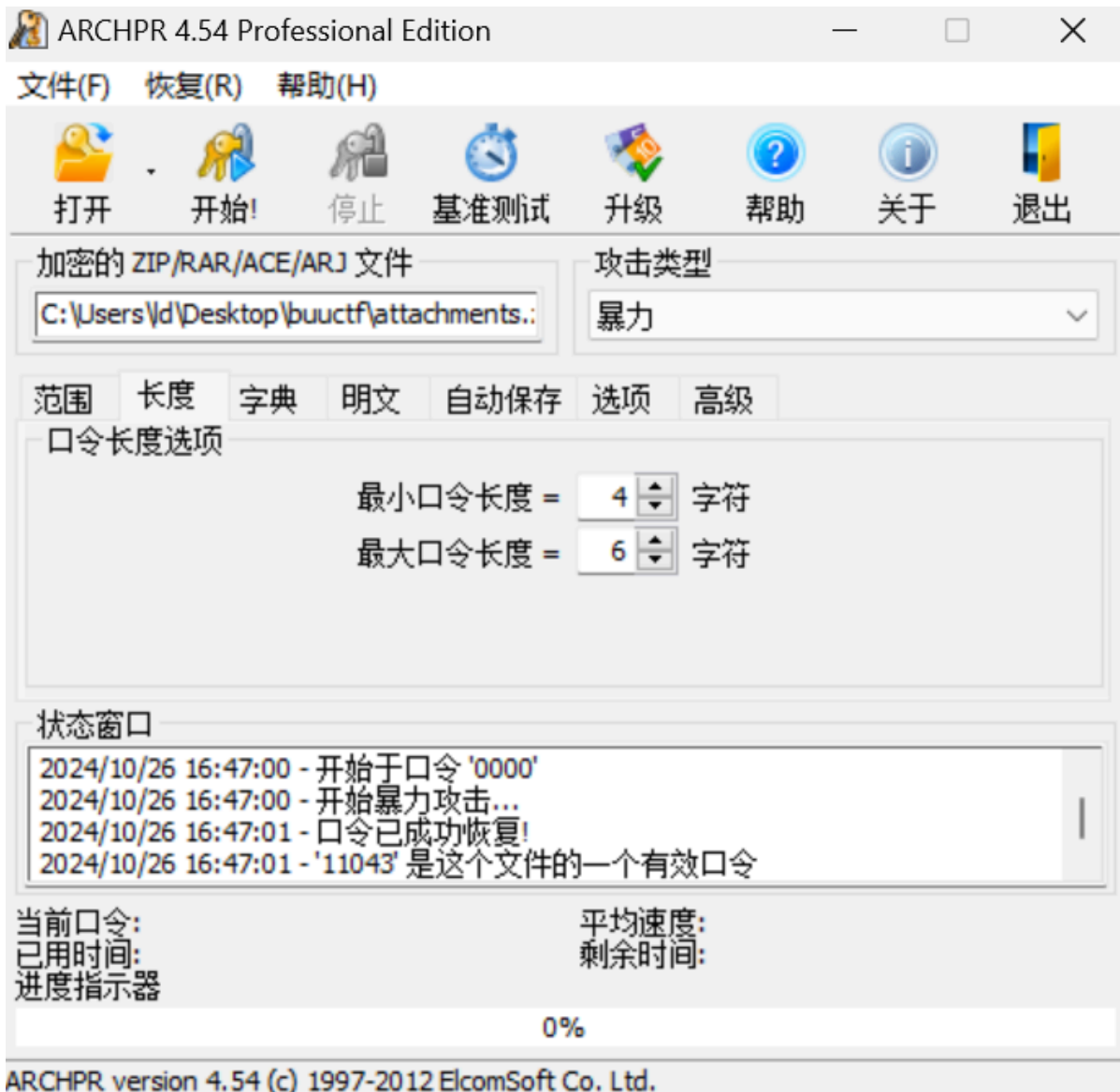
我们就将attachment.jpg后缀名改为.zip将其打回原形

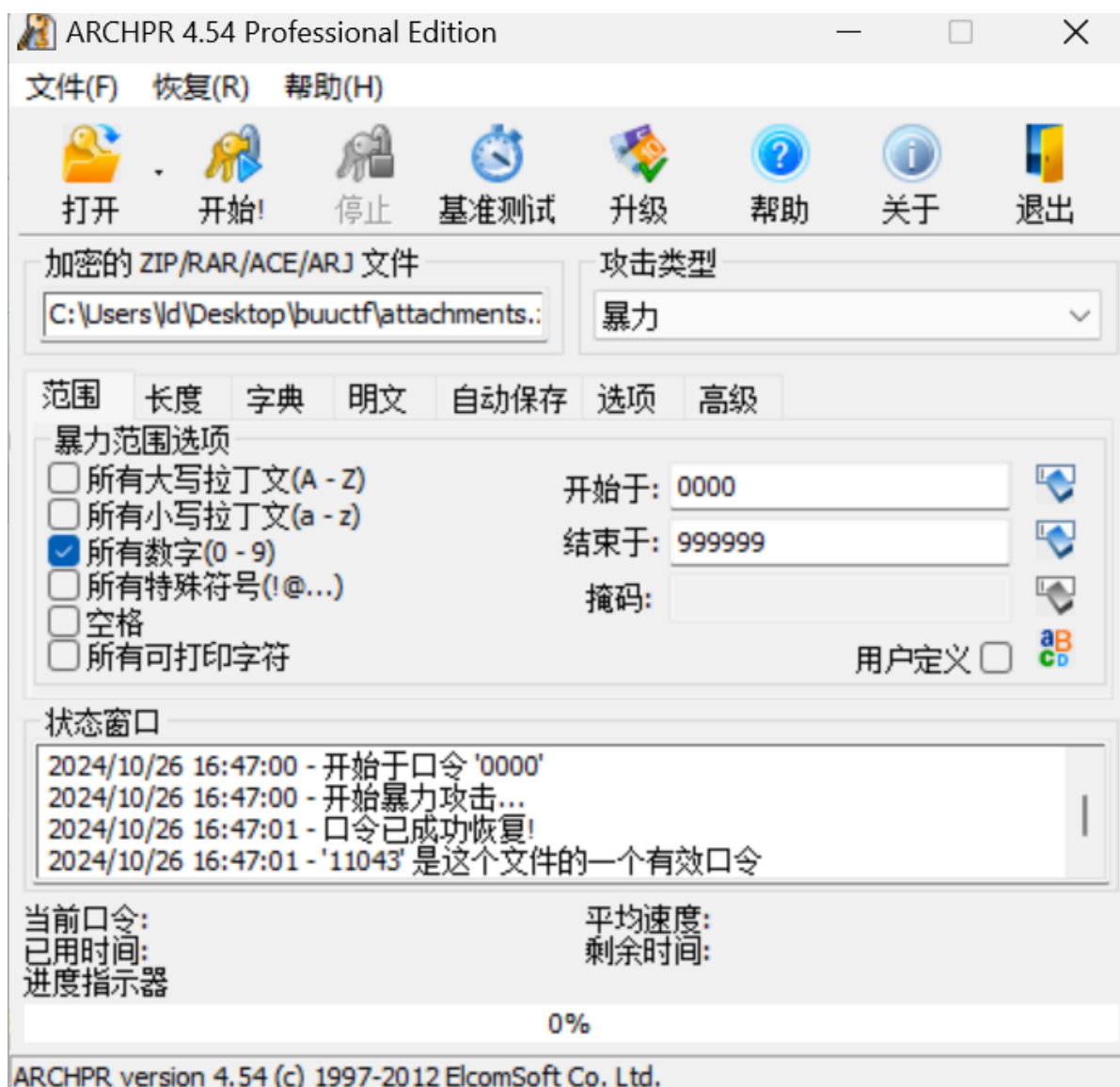
 attachments.zip	2024/9/27 16:04	ZIP 压缩文件	1 KB
---	-----------------	----------	------

解压后应该就有flag了



纳尼，居然还要密码，那我们就尝试一下密码爆破，这里我们使用ARCHPR暴力爆破一下（设置如下）





输入密码以后获得flag.txt

