**伪加密**

1. **ZIP伪加密**

#### **原理**

在上文 ZIP 格式中的 **核心目录区** 中，我们强调了一个叫做通用位标记 (General purpose bit flag) 的 2 字节，不同比特位有着不同的含义。

Bit 0: If set, indicates that the file is encrypted.

(For Method 6 - Imploding)

Bit 1: If the compression method used was type 6,

Imploding, then this bit, if set, indicates

an 8K sliding dictionary was used. If clear,

then a 4K sliding dictionary was used.

...

Bit 6: Strong encryption. If this bit is set, you should

set the version needed to extract value to at least

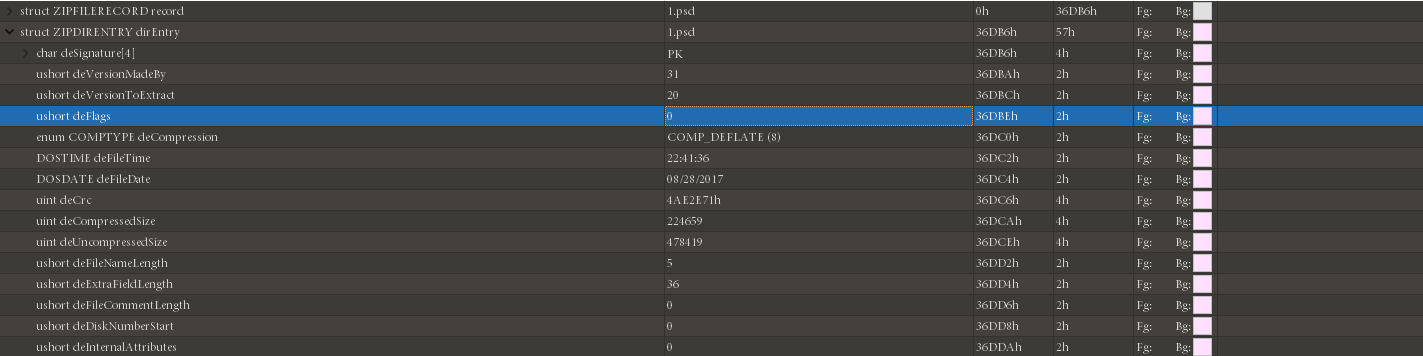
50 and you must also set bit 0. If AES encryption

is used, the version needed to extract value must

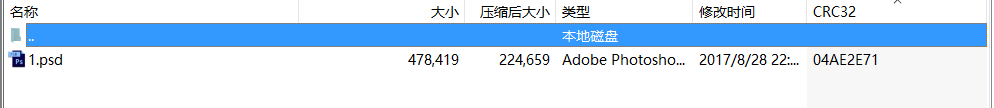
be at least 51.

...

在 010Editor 中我们尝试着将这 1 位修改 0 --> 1。



再打开文件发现已要求输入密码。



**修改伪加密的方法：**

1. 16 进制下修改通用位标记
2. binwalk -e 无视伪加密
3. 在 Mac OS 及部分 Linux(如 Kali ) 系统中，可以直接打开伪加密的 ZIP 压缩包
4. 检测伪加密的小工具 ZipCenOp.jar
5. 有时候用 WinRar 的修复功能（此方法有时有奇效，不仅针对伪加密）

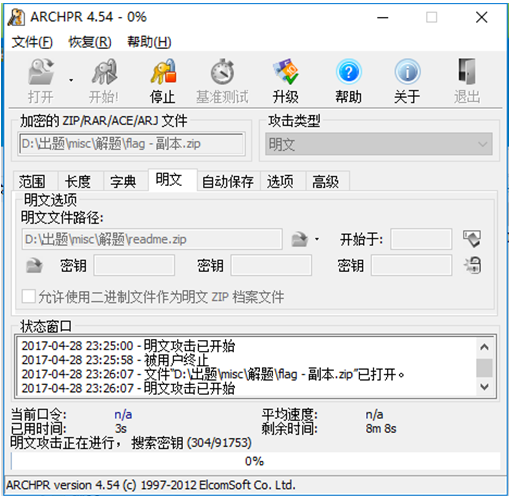
#### **例题**

SSCTF-2017 ：我们的秘密是绿色的

WP：<http://bobao.360.cn/ctf/detail/197.html>

我们在得到两个 readme.txt，且一个加密，一个已知，很容易想到明文攻击的手法。

注意在用明文攻击时的操作。



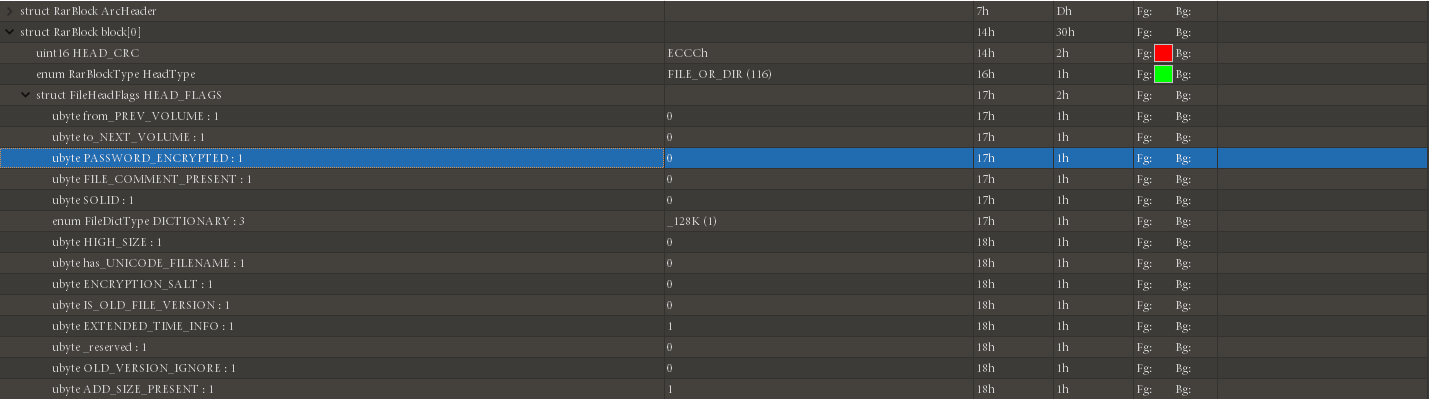
得到密码 Y29mZmVl 后，解压缩文件，得到另一个压缩包。

观察通用位标记位，猜测伪加密，修改后解压得到 flag。

这一题，基本涵盖了比赛中 ZIP 的常见考察手法，爆破，伪加密，明文攻击等，都在本题中出现。

1. RAR伪加密

RAR 文件的伪加密在文件头中的位标记字段上，用 010 Editor 可以很清楚的看见这一位，修改这一位可以造成伪加密。



其余明文攻击等手法依旧同 ZIP 中介绍的一样。