

# ctf隐写篇

原创

matrix\_123 于 2017-10-14 17:06:34 发布 8490 收藏 18

分类专栏: [ctf](#) 文章标签: [隐写](#) [图片](#) [密码学](#) [安全](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: [https://blog.csdn.net/Life\\_hes\\_az/article/details/78235435](https://blog.csdn.net/Life_hes_az/article/details/78235435)

版权



[ctf 专栏收录该内容](#)

1 篇文章 0 订阅

订阅专栏

本人从大学开始接触ctf, 想对ctf中的隐写题做一个总结。

隐写其实是一门很深的学问, 在ctf中我们常见的是图片的隐写, 以下我们将谈到的是一些基本的图片隐写及其所涉及的一些问题。

## 一个概念

隐写指的是隐藏信息, 信息是不可见的, 而在密码学中, 信息是可见的, 只是顺序被打乱了, 直接观察让人无法理解。

## 一、图种

图种主要是对于JPG(一种有损压缩)格式的图片来说的, 利用了图片查看器的一个特点, 查看器只检查jpg图片的头(FF D8), 不会去检查图片尾(FF D9)及其以后的内容。图种就是将隐藏文件加在了FFD9后面

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
00000000	FF	D8	FF	E0	00	10	4A	46	49	46	00	01	01	01	00	B4	ÿøÿà JFIF
00000010	00	B4	00	00	FF	DB	00	43	00	06	04	05	06	05	04	06	ÿÜ C
00054AA0	C5	7A	1F	6A	7F	EE	6D	FF	00	D4	1F	95	61	35	1F	EE	Àz j imÿ Ô •a5 î
00054AB0	6A	B1	76	77	43	23	5D	1F	FF	D9							j±vwC#j ÿÜ

图种的制作其实很简单, cmd中执行命令 `copy /b 1.jpg+2.zip out.jpg`

```
C:\Users\...\Desktop>copy /b 1.jpg+2.zip out.jpg
1.jpg
2.zip
已复制      1 个文件
```

发现FFD9后多了一些内容

6A	B1	76	77	43	23	5D	1F	FF	D9	50	4B	03	04	14	00		j±vwC#j ÿÜPK
00	00	08	00	C7	9E	50	4B	72	BC	20	DD	A1	00	00	00		ÇžPKr±&Ý;
4B	01	00	00	0C	00	19	00	BD	E2	C3	DC	CF	B5	C1	D0		K ±áÄÜÿ±Ä
2E	74	78	74	75	70	15	00	01	02	5A	67	AE	E8	A7	A3		±.txtup ZgSèS£
E5	AF	86	E7	B3	BB	E5	88	97	2E	74	78	74	3B	7C	E7		ä±tç³»ä±.txt; ç

当在ctf中遇到一个jpg格式的图片，用winhex打开它，使用winhex的快捷键ctrl+alt+x搜索一下找到FFD9  
看一下其后有没有其他的内容，如果后面发现还有其他信息（通常是 50 4B），如果是50 4B，可以直接将文件后缀名改为zip，  
就可以得到隐写的文件，或者使用两款工具binwalk和foremost，binwalk会去检查文件的格式发现其中是否含有其他文  
件，foremost主要的作用是将含有的其他文件分离出来。

```
root@kali:~# cd Desktop/
root@kali:~/Desktop# ls
out.jpg
root@kali:~/Desktop# binwalk out.jpg
DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0            0x0          JPEG image data, JFIF standard 1.01
347157      0x54C15     End of Zip archive

root@kali:~/Desktop# foremost out.jpg
Processing: out.jpg
| foundat=0000e00.txtup[0]
*|
http://blog.csdn.net/Life_hes_az
```

在此目录先生成output文件夹

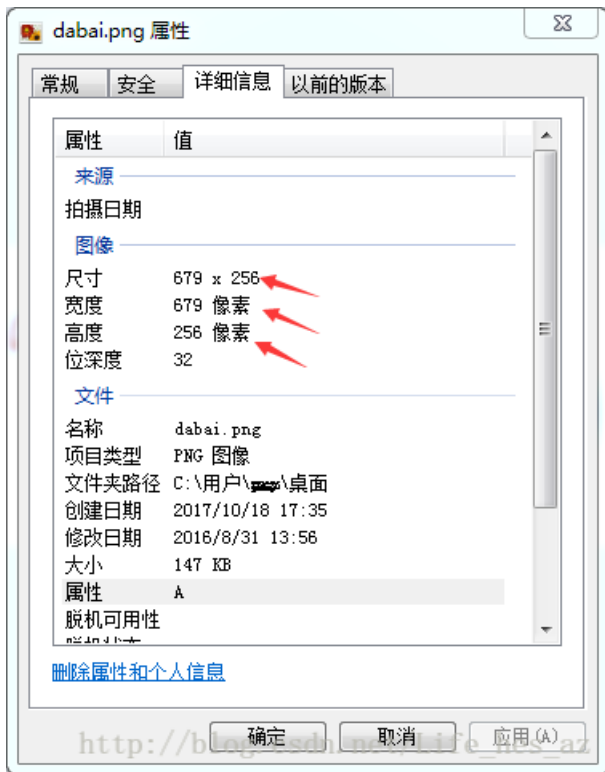
其实只要了解文件格式，图种其实很简单，在ctf题中只作为考察的一部分，通常会与其他内容相结合来考察。

## 二、png

png（无损压缩格式）的详细格式请见[链接](#)

一道ctf题：[链接：http://pan.baidu.com/s/1i5GfgA5](http://pan.baidu.com/s/1i5GfgA5) 密码：v47e

看到是一个png的图片，先用图片查看器打开图片，查看详细信息



发现图片的高度和宽度不太对，winhex打开

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
00000000	89	50	4E	0D	0A	1A	0A	00	00	00	0D	49	48	44	52		%PNG IHDR
00000010	00	00	02	A7	00	00	01	00	08	06	00	00	00	6D	7C	71	\$ m q
00000020	35	00	00	00	01	78	52	47	42	00	AE	CE	1C	E9	00	00	5 sRGB @I é
00000030	00	04	67	41	4D	41	00	00	B1	8F	0B	FC	61	05	00	00	gAMA ± ũa
00000040	00	09	70	48	59	73	00	00	0E	C4	00	00	0E	C4	01	95	pHYs Ä Ä ·
00000050	2B	0E	1B	00	00	FF	A5	49	44	41	54	78	5E	EC	BD	07	+ ŷIDATx^i%
00000060	A0	A5	57	59	EE	FF	EE	BE	4F	9B	DE	93	4C	7A	0F	84	ŷWYiŷi%>P`Lz`
00000070	24	24	60	0C	04	A5	2B	20	45	10	10	BB	88	8A	A8	57	\$\$ ŷ+ E » S W

对格式做一下解释

```

89 50 4E 0D 0A 1A 0A //图像的标识
00 00 00 0D //文件头数据块IHDR的长度 0x0D = 13Byte
49 48 44 52 //IHDR标识
00 00 02 A7 //图像的宽度, 单位为像素 4byte
00 00 01 00 //图像的高度, 单位为像素 4byte

```

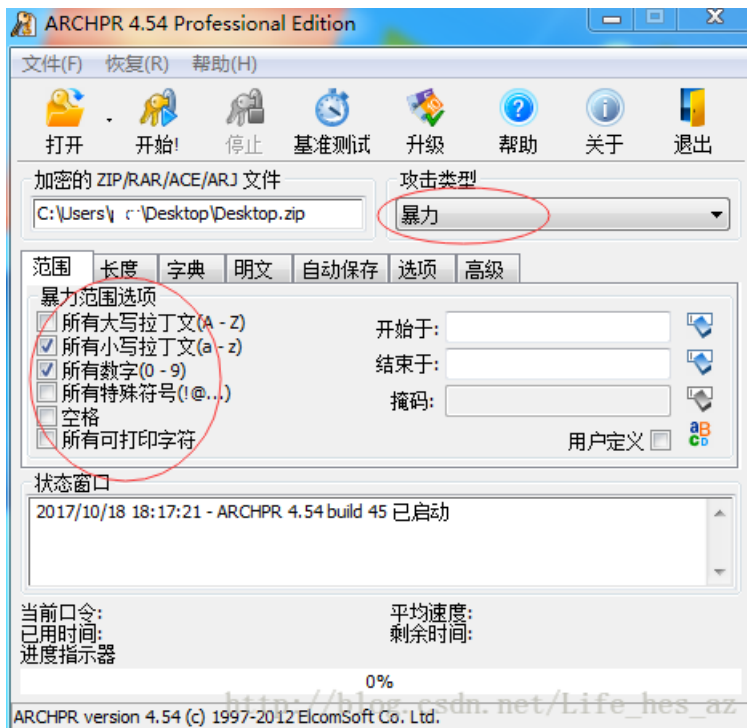
后面还有5Byte的内容为图片的颜色方面的内容  
修改图片的高度, 得到flag.

### 三、zip

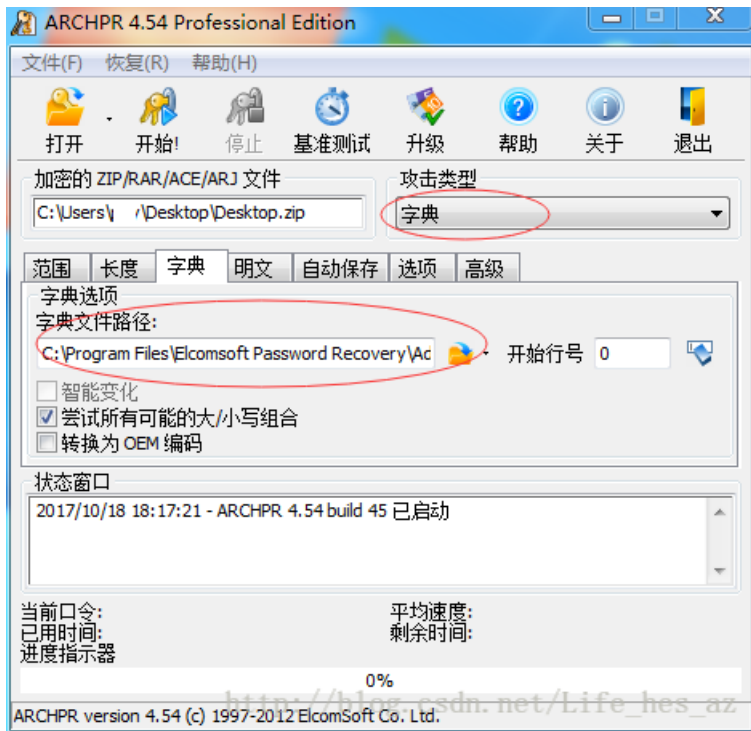
对于zip文件通常有爆破、字典、伪加密、掩码攻击、明文攻击等, 主要是使用工具AZPR

伪加密: 主要是与zip的格式有关, 要了解zip文件的格式. 其主要与zip的加密位有关, 可以参考[链接](#)

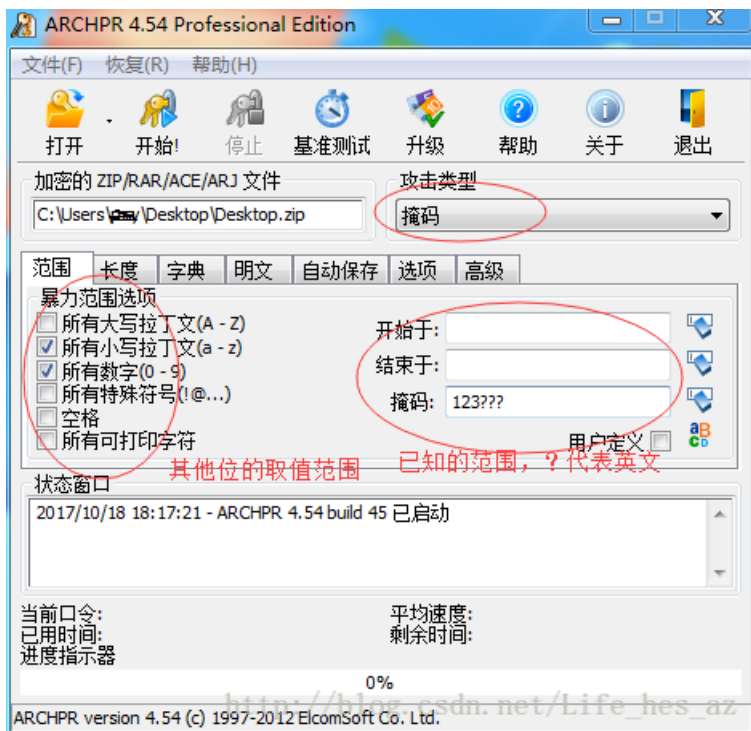
爆破: 即暴力破解, 利用ZipRar进行爆破, 这种方法难度较大, 成功率较低



字典: 通过字典进行爆破, 通常可以根据题目的提示, 如生日啦. 来选取相应的字典



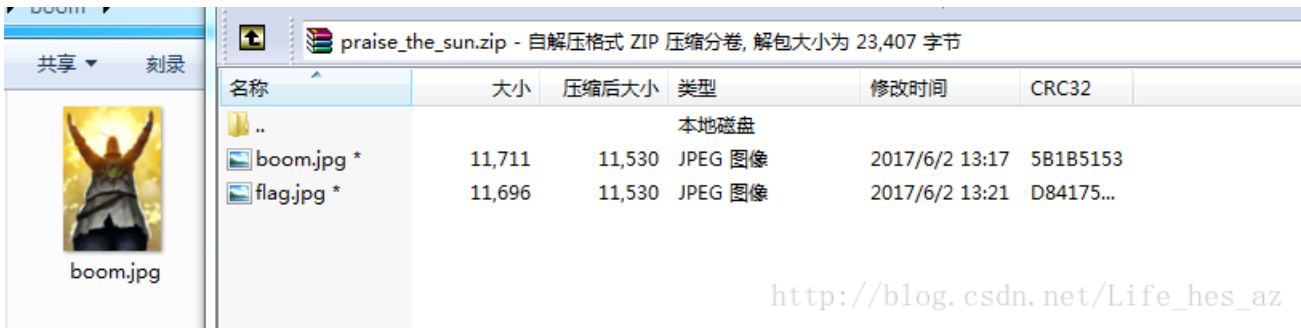
掩码攻击：根据题目知道了密码的范围或者是得到密码的部分，来猜其他位。



明文攻击：这是一种较为高效的方法，对于一个zip加密文件，已知加密文件中的某一个文件，将已知文件按加密文件的压缩算法压缩，在使用明文攻击，可以较快得到结果。明文攻击主要的原理是利用了两文件的压缩算法相同，用已知文件去撞加密文件中的已知文件。

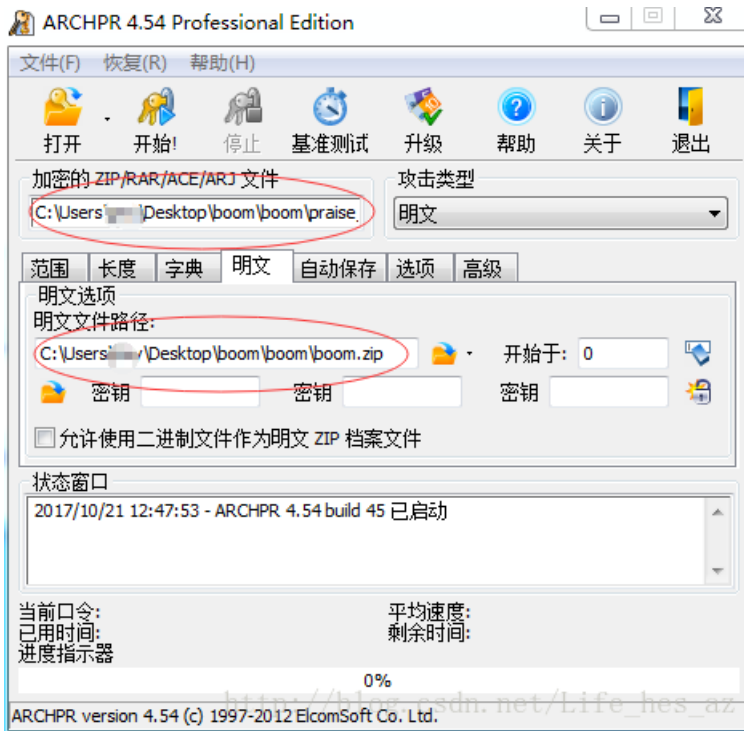
链接：<http://pan.baidu.com/s/1dFrQJPR> 密码：wxyp

打开压缩包，检查文件（binwalk），发现存在图种，用foremost分离，遇到zip加密，通过观察发现加密zip中有文件为已知文件

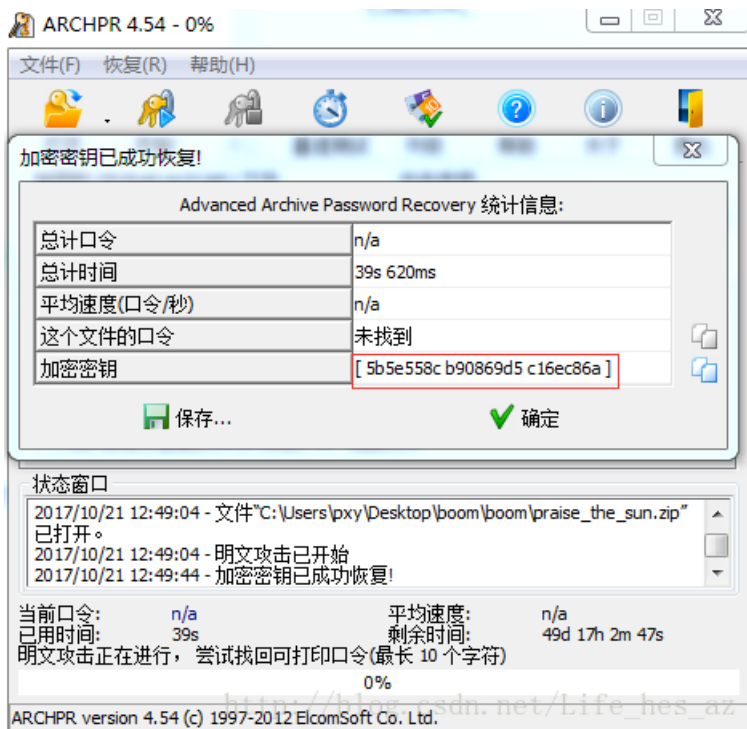


[http://blog.csdn.net/Life\\_hes\\_az](http://blog.csdn.net/Life_hes_az)

发现可以进行明文攻击



[http://blog.csdn.net/Life\\_hes\\_az](http://blog.csdn.net/Life_hes_az)



最好得到三个key，虽没有爆出密码但是得到了加密zip中的文件（flag.jpg），检查一下，发现了不为人知的秘密。

### crc32爆破

crc(Cyclic Redundancy Check)是一种循环冗余校验码，主要是数据传输过程中用来检错的，也可称为检错码，主要是原理是利用了二进制的模2运算，在这里就不展开了

主要的思路

这个主要是要写脚本，去爆破它。

## 四、data:image/类型 + ntfs文件流 +盲水印隐写

### data:image/类型

```
<img src="" /> //将文件内容放入""中，将文件后缀名改为html
```

用浏览器打开就见到图片了

或者的使用在线的解码网站

### ntfs文件流

#### 盲水印隐写

一道ctf题链接：<http://pan.baidu.com/s/1sIDhvf> 密码：vroj

首先是使用明文攻击，得到加密压缩包中的图片，发现fuli.png和fuli2.png两个文件，判断是盲水印

脚本

使用 bwm.py

```
python bwm.py decode fuli.png fuli2.png wm_out.png
```

得到flag

通过一些了解，我们发现在ctf比赛中，对文件头了解是很重要的

各种文件头格式<http://www.cnblogs.com/13ck/p/4471146.html>

总结一下

以上所说的一些方法，在ctf比赛中往往是多个套路融合在一起出题，当然上面的也只是一些常见的套路，我了解的还是很少，更多的主要是自己多去练习了，平时多去了解一些文件的结构。

推荐一本书[数据隐藏技术揭秘：破解多媒体、操作系统、移动设备和网络协议中的隐秘数据](#)这本书说的很全面，值得去阅读一下。