CTF压缩包隐写类（zip、RAR、zip伪加密）

原创

Hardworking666 于 2022-01-01 12:18:12 发布

分类专栏： CTF 文章标签： zip RAR CTF 压缩包 隐写

本文链接：https://blog.csdn.net/Hardworking666/article/details/122266479

文章目录

一、zip

CTF中的压缩包隐写一般有这样几个套路

1、通过编码转换隐藏信息（common）
比如给出一堆字符或数字，仔细观察为某种进制，将其解码为十六进制，观察其文件头是压缩包或者是其他格式，修改后缀名后解压得flag

2、在文件中隐藏压缩包（图种）
在CTF压缩包隐写中最为常见，多用于在一个文件中隐藏一个压缩包

原理：以jpg格式为例，完整的JPG由FF D8开头，FF D9结束，图片浏览器会忽略FF D9之后的内容，因此可以在JPG文件之后加入其他的文件。

利用foremost，dd或者直接将其修改为压缩包后缀进行提取。
推荐使用foremost，因为foremost还可以分离其他隐藏的文件。
修改为ZIP文件虽然方法简单，但是如果隐写了多个文件时可能会失败。
以前不知道foremost的时候一直是用dd分离的，后边知道了foremost就一直用的foremost。

3、伪加密
原理：ZIP伪加密是在文件头的加密标志位进行修改，进而再次打开文件时被识别为加密压缩包。

ZIP文件主要由三个部分组成：压缩源文件数据区 + 核心目录 + 目录结束标志

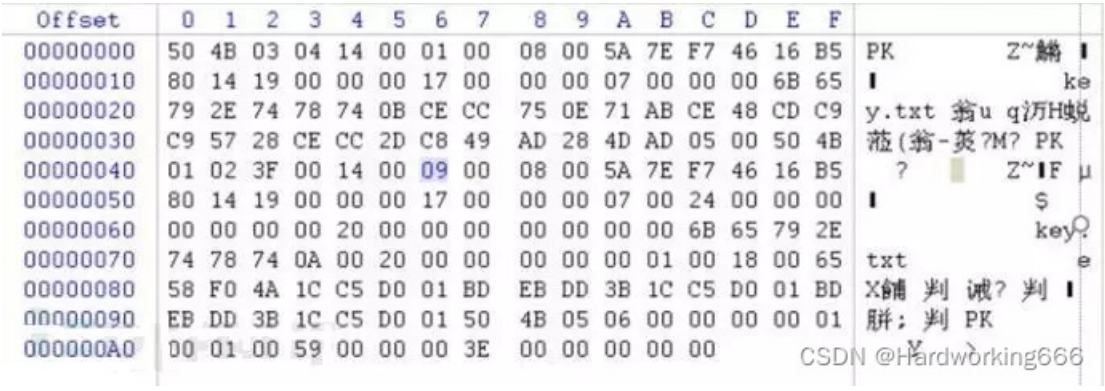
压缩源文件数据区

local file header + file data + data descriptor

```
local file header：文件头用于标识该文件的开始，记录了该压缩文件的信息，
这里的文件头标识由固定值 50 4B 03 04 开头，也是 ZIP 的文件头的重要标志。

file data：文件数据记录了相应压缩文件的数据。

data descriptor：数据描述符用于标识该文件压缩结束，
该结构只有在相应的 local file header 中通用标记字段的第 3 bit设为 1 时才会出现，
紧接在压缩文件源数据后。
```

```
50 4B 03 04：这是头文件标记（0x04034b50）
14 00：解压文件所需 pkware 版本
00 00：全局方式位标记（有无加密）
08 00：压缩方式
5A 7E：最后修改文件时间
F7 46：最后修改文件日期
16 B5 80 14：CRC-32校验（1480B516）
19 00 00 00：压缩后尺寸（25）
17 00 00 00：未压缩尺寸（23）
07 00：文件名长度
00 00：扩展记录长度
```

6B65792E7478740BCECC750E71ABCE48CDC9C95728CECC2DC849AD284DAD0500 （直到核心目录文件头标识）



压缩源文件目录区

记录了压缩文件的目录信息，在这个数据区中每一条纪录对应在压缩源文件数据区中的一条数据。

```
50 4B 01 02：目录中文件文件头标记(0x02014b50)
3F 00：压缩使用的 pkware 版本
14 00：解压文件所需 pkware 版本
00 00：全局方式位标记（有无加密，这个更改这里进行伪加密，改为09 00打开就会提示有密码了）
08 00：压缩方式
5A 7E：最后修改文件时间
F7 46：最后修改文件日期

16 B5 80 14：CRC-32校验（1480B516）
19 00 00 00：压缩后尺寸（25）
17 00 00 00：未压缩尺寸（23）
07 00：文件名长度
24 00：扩展字段长度
00 00：文件注释长度
00 00：磁盘开始号
00 00：内部文件属性
20 00 00 00：外部文件属性
00 00 00 00：局部头部偏移量
6B65792E7478740A002000000000000010018006558F04A1CC5D001BDEBDD3B1CC5D001BDEBDD3B1CC5D001（直到目录结束标识头）
```

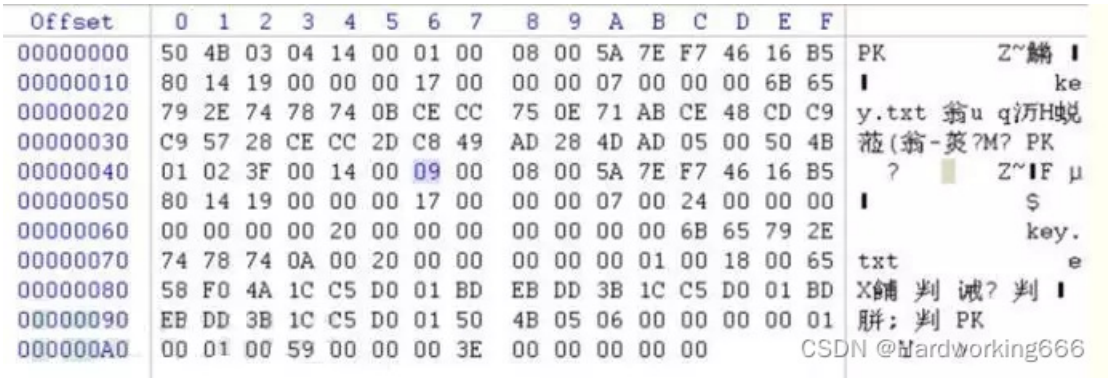
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	50	4B	03	04	14	00	01	00	08	00	5A	7E	F7	46	16	B5	PK Z~鱗 ▮
00000010	80	14	19	00	00	00	17	00	00	00	07	00	00	00	6B	65	▮ ke
00000020	79	2E	74	78	74	0B	CE	CC	75	0E	71	AB	CE	48	CD	C9	y.txt 鈥u q汚H蜕
00000030	C9	57	28	CE	CC	2D	C8	49	AD	28	4D	AD	05	00	50	4B	蒞(鈥-英?M? PK
00000040	01	02	3F	00	14	00	09	00	08	00	5A	7E	F7	46	16	B5	? Z~▮F µ
00000050	80	14	19	00	00	00	17	00	00	00	07	00	24	00	00	00	▮ $
00000060	00	00	00	00	20	00	00	00	00	00	00	00	6B	65	79	2E	key.
00000070	74	78	74	0A	00	20	00	00	00	00	00	01	00	18	00	65	txt e
00000080	58	F0	4A	1C	C5	D0	01	BD	EB	DD	3B	1C	C5	D0	01	BD	X鋪 判 诫? 判 ▮
00000090	EB	DD	3B	1C	C5	D0	01	50	4B	05	06	00	00	00	00	01	胼; 判 PK
000000A0	00	01	00	59	00	00	00	3E	00	00	00	00	00				


```
Central directory structure:

[file header 1]
.
.
.
[file header n]
[digital signature]

File header:

  central file header signature   4 bytes  (0x02014b50)
  version made by                 2 bytes
  version needed to extract       2 bytes
  general purpose bit flag        2 bytes
  compression method              2 bytes
  last mod file time              2 bytes
  last mod file date              2 bytes
  crc-32                          4 bytes
  compressed size                 4 bytes
  uncompressed size               4 bytes
  file name length                2 bytes
  extra field length              2 bytes
  file comment length             2 bytes
  disk number start               2 bytes
  internal file attributes        2 bytes
  external file attributes        4 bytes
  relative offset of local header 4 bytes

  file name (variable size)
  extra field (variable size)
  file comment (variable size)
```

目录结束标识（End of Central Directory Record)

存在于整个归档包的结尾，用于标记压缩的目录数据的结束。每个压缩文件必须有且只有一个结束标识。

```
50 4B 05 06：目录结束标记
00 00：当前磁盘编号
00 00：目录区开始磁盘编号
01 00：本磁盘上纪录总数
01 00：目录区中纪录总数
59 00 00 00：目录区尺寸大小
3E 00 00 00：目录区对第一张磁盘的偏移量
00 00：ZIP 文件注释长度
```

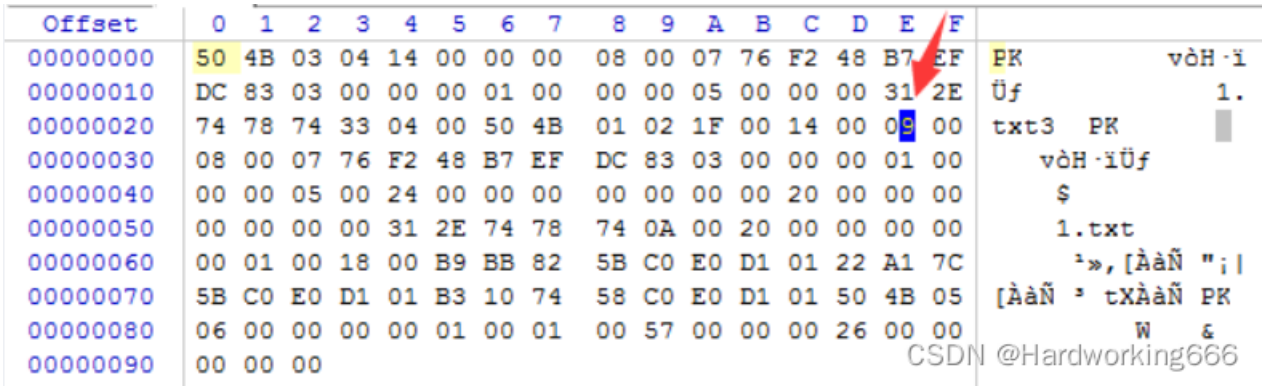


Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	50	4B	03	04	14	00	01	00	08	00	5A	7E	F7	46	16	B5
00000010	80	14	19	00	00	00	17	00	00	00	07	00	00	00	6B	65
00000020	79	2E	74	78	74	0B	CE	CC	75	0E	71	AB	CE	48	CD	C9
00000030	C9	57	28	CE	CC	2D	C8	49	AD	28	4D	AD	05	00	50	4B
00000040	01	02	3F	00	14	00	09	00	08	00	5A	7E	F7	46	16	B5
00000050	80	14	19	00	00	00	17	00	00	00	07	00	24	00	00	00
00000060	00	00	00	00	20	00	00	00	00	00	00	00	6B	65	79	2E
00000070	74	78	74	0A	00	20	00	00	00	00	00	01	00	18	00	65
00000080	58	F0	4A	1C	C5	D0	01	BD	EB	DD	3B	1C	C5	D0	01	BD
00000090	EB	DD	3B	1C	C5	D0	01	50	4B	05	06	00	00	00	00	01
000000A0	00	01	00	59	00	00	00	3E	00	00	00	00	00			

zip伪加密

zip伪加密是在文件头的加密标志位做修改，进而再打开文件时识被别为加密压缩包。

如果把第二个**加密标记位**的00 00改为09 00，打开就会提示有密码：



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	50	4B	03	04	14	00	00	00	08	00	07	76	F2	48	B7	EF
00000010	DC	83	03	00	00	00	01	00	00	00	05	00	00	00	31	2E
00000020	74	78	74	33	04	00	50	4B	01	02	1F	00	14	00	09	00
00000030	08	00	07	76	F2	48	B7	EF	DC	83	03	00	00	00	01	00
00000040	00	00	05	00	24	00	00	00	00	00	00	00	20	00	00	00
00000050	00	00	00	00	31	2E	74	78	74	0A	00	20	00	00	00	00
00000060	00	01	00	18	00	B9	BB	82	5B	C0	E0	D1	01	22	A1	7C
00000070	5B	C0	E0	D1	01	B3	10	74	58	C0	E0	D1	01	50	4B	05
00000080	06	00	00	00	00	01	00	01	00	57	00	00	00	26	00	00
00000090	00	00	00													

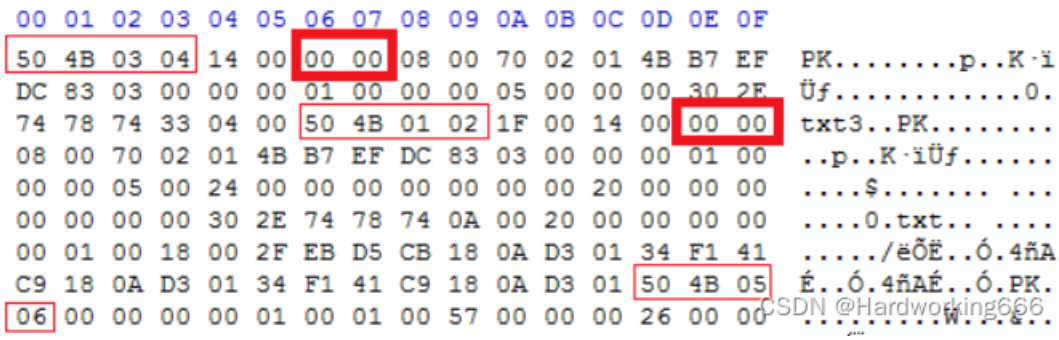
其实改成**09**只是举的一个例子，只要末位是**奇数**，就代表加密，反之，末位是偶数代表未加密。
有时这里是**01**，也代表加密！不用更改！

识别真假加密

无加密
压缩源文件数据区的全局加密应当为00 00
且压缩源文件目录区的全局方式位标记应当为00 00

假加密
压缩源文件数据区的全局加密应当为00 00
且压缩源文件**目录**区的全局方式位标记应当为09 00

真加密
压缩源文件数据区的全局加密应当为09 00
且压缩源文件目录区的全局方式位标记应当为09 00



二、RAR

文件格式

RAR 文件主要由标记块，压缩文件头块，文件头块，结尾块组成。

其每一块大致分为以下几个字段：

名称	大小	描述
HEAD_CRC	2	全部块或块部分的CRC
HEAD_TYPE	1	块类型
HEAD_FLAGS	2	阻止标志
HEAD_SIZE	2	块大小
ADD_SIZE	4	可选字段 - 添加块大小

RAR压缩包的文件头为：52 61 72 21 1A 07 00

其后是标记块（MARK_HEAD），还有文件头（FILE_HEAD）。

更多信息见：http://www.forensicswiki.org/wiki/RAR

主要攻击方式

1、爆破

利用linux下的rarcrack（http://rarcrack.sourceforge.net/）

2、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。

3、其他如明文攻击等方法与ZIP相同。