

CTF 关于ZIP解题过程

原创

Sn0w/ 于 2019-03-31 18:06:55 发布 4162 收藏 12

分类专栏: [CTF_Writeup](#) 文章标签: [ctf zip](#)

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/qq_43431158/article/details/88781970

版权



[CTF_Writeup](#) 专栏收录该内容

32 篇文章 4 订阅

订阅专栏

CTF 关于ZIP解题

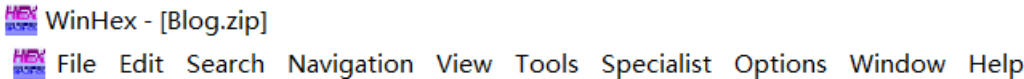
1.伪加密类型

Blog.zip	2019/3/24 20:40	WinRAR ZIP 压缩...	390 KB
----------	-----------------	------------------	--------

打开之后是两张图片, 一张未加密, 一张已加密。

名称	大小	压缩后大小	类型	修改时间	CRC32
..			本地磁盘		
b.png *	207,670	204,652	PNG 文件	2019/3/21 20:...	0268F447
a.png	226,792	193,603	PNG 文件	2019/3/21 21:...	E3197921

利用WinHex打开



然后在谷歌上找到大佬关于zip格式的介绍

b.压缩源文件目录区:

- 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

6B65792E7478740A00200000000000010018006558F04A1CC5D001BDEBDD3B1CC5D001BDEBDD3B1CC5D001

https://blog.csdn.net/qq_43431158

接下来进行对比，看是否属于伪加密

000613F0	1B FF A4 7F 00 50 4B 01 02 3F 00 14 00 00 00 08	ÿ PK ?
00061400	00 9A A8 75 4E 21 79 19 E3 43 F4 02 00 E8 75 03	š uN!y äCò èu
00061410	00 05 00 24 00 00 00 00 00 00 00 20 00 00 00 00	\$
00061420	00 00 00 01 2E 70 6E 67 0A 00 20 00 00 00 00 00	.png
00061430	01 00 18 00 F1 EB A4 AB E6 DF D4 01 4B 2B 72 AB	ñè««æßÔ K+r«
00061440	E6 DF D4 01 7F 14 29 AB E6 DF D4 01 50 4B 01 02	æßÔ)«æßÔ PK
00061450	3F 00 14 00 09 00 08 00 15 A6 75 4E 47 F4 68 02	? !uNGôh
00061460	6C 1F 03 00 36 2B 03 00 05 00 24 00 00 00 00 00	l 6+ \$
00061470	00 00 20 00 00 00 66 F4 02 00 62 2E 70 6E 67 0A	fô b.png
00061480	00 20 00 00 00 00 00 01 00 18 00 62 8D E7 68 E4	b çhä
00061490	DF D4 01 2F 10 E2 68 E4 DF D4 01 6A 72 8A 8B E1	ßÔ / âhãßÔ jrŠ<á
000614A0	DF D4 01 50 4B 05 06 00 00 00 00 02 00 02 00 AE	ßÔ PK
000614B0	00 00 00 F5 13 06 00 00 00	õ

通过对比，发现确实属于伪加密。所以将09改为00即可。

总结：从50 4B 01 02 开始数十位数便能查看是否是伪加密。

CRC是个校验码

10

这个题是CRC32碰撞，首先有不会的就百度、谷歌。
通过查找发现需要CRC32碰撞的脚本和安装pathon。

一：安装CRC32 Tools

📖 README.md

CRC32 Tools



工具地址

二：安装pathon

[Python_官方电脑版_华军纯净下载](#)



版本 : 3.7.2 for Windows
大小 : 24.19MB
更新 : 2019-02-12
环境 : WinAll

立即下载

https://blog.csdn.net/qq_43431158

因为要翻墙，去官网下载会特别慢，下载这个没有病毒，也可以用。
还有下载好之后，最好找一下这个版本的安装教程。

[安装3.7.2pathon的教程](#)

配置好之后，打开cmd，进入你安装脚本的目录。

```
D:\>cd crc32
```

```
D:\crc32>
```

打开下载好的压缩包：

名称	大小	压缩后大小	类型	修改时间	CRC32
..			本地磁盘		
2.txt *	6	18	文本文档	2019/3/19 20:...	127F1984
3.txt *	6	18	文本文档	2019/3/19 20:...	4FA83D...
flag.txt *	102	114	文本文档	2019/3/19 20:...	7F01762D
1.txt *	6	18	文本文档	2019/3/19 20:...	7F616EE3

记录下1.2.3的CRC32值

利用碰撞的语法：

```
python crc32.py reverse "你的crc32密文"
```

```
D:\crc32>python crc32.py reverse "0X7F616EE3"
```

```
4 bytes: {0xfc, 0xf3, 0x48, 0x10}
```

```
verification checksum: 0x7f616ee3 (OK)
```

```
alternative: 06iBmA (OK)
```

```
alternative: 2GAaYT (OK)
```

```
alternative: 4BH2ir (OK)
```

```
alternative: 8LCPd9 (OK)
```

```
alternative: AGtKKP (OK)
```

```
alternative: Dbvk8f (OK)
```

```
alternative: ECiJJ3 (OK)
```

```
alternative: Hp2U49 (OK)
```

```
alternative: M9CXCK (OK)
```

```
alternative: TrCiM1 (OK)
```

```
alternative: WoYVfy (OK)
```

```
alternative: _e05jQ (OK)
```

```
alternative: aHGrpU (OK)
```

```
alternative: eLZsq6 (OK)
```

```
alternative: k3y2Hh (OK)
```

```
alternative: kCECM8 (OK)
```

```
alternative: l61PcW (OK)
```

```
alternative: m6paxN (OK)
```

```
alternative: pymQwW (OK)
```

```
alternative: xo4nzk (OK)
```

```
alternative: you_ar (OK)
```

```
<!--破解1.flag-->
```

```
D:\crc32>python crc32.py reverse "0x127F1984"
4 bytes: {0x0c, 0xa9, 0xe2, 0xfd}
verification checksum: 0x127f1984 (OK)
alternative: 1IuEfu (OK)
alternative: 7P3JWG (OK)
alternative: 8_mKpP (OK)
alternative: ATZP_9 (OK)
alternative: K_XabT (OK)
alternative: MZQ2Rr (OK)
alternative: 076Mgs (OK)
alternative: SxjLss (OK)
alternative: TamrYX (OK)
alternative: ZnrBeV (OK)
alternative: bFsV0t (OK)
alternative: cF2gTm (OK)
alternative: e_the_ (OK)
alternative: kPkXYQ (OK)
alternative: lIlfsz (OK)
alternative: n8DEGo (OK)
alternative: oTvYX2 (OK)
alternative: swYuHv (OK)
<!--破解2.flag-->
```

```
D:\crc32>python crc32.py reverse "0x4FA83D8C"
4 bytes: {0x7e, 0xfa, 0xeb, 0x0a}
verification checksum: 0x4fa83d8c (OK)
alternative: 0KjFzu (OK)
alternative: 3ka59e (OK)
alternative: AwZr1l (OK)
alternative: CK_1hq (OK)
alternative: DRXRbZ (OK)
alternative: LXN1Nr (OK)
alternative: PFpQ6n (OK)
alternative: RzuOos (OK)
alternative: UcrqEX (OK)
alternative: a5Dvga (OK)
alternative: bXb8Iy (OK)
alternative: cDlUSt (OK)
alternative: lK2Ttc (OK)
alternative: mKseoz (OK)
alternative: nViZD2 (OK)
alternative: ruFvTv (OK)
<!--破解3.flag-->
```

接下来，找到有意义或连贯的的英语词语。

通过查找发现有几个有意义的词语。

```
e_the_
you_ar
```

一开始以为答案就应该是：flag{you_are_the_}，但提交还是错误。

通过查找，发现漏了一种可能，就是特殊字符。

CRC32和MD5加密都是不可逆的，也就是说加密之后是不能反过来看我之前的内容。所以有的大佬就将常见的词（例如A,B,C。。。）写成了一个脚本,通过一个一个对比来破解出之前的密文。

打开我们下载的CRC32脚本。

```
permitted_characters = set(  
    map(ord, 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789_')) # \w
```

上面便是大佬脚本中的常见词，接下来我们自己添加一些特殊字符。

```
permitted_characters = set(  
    map(ord, 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789! , . ; 0 ')) # \w
```

可以看到我自己添加了；/，！等这些特殊符号。（还有一点不要在最后面加特殊字符，会有问题，最好在0前面加）。

再次破解1.2.3.flag

发现3中出现有意义的词语。

```
D:\crc32>python crc32.py reverse "0x4FA83D8C"  
4 bytes: {0x7e, 0xfa, 0xeb, 0x0a}  
verification checksum: 0x4fa83d8c (OK)  
alternative: ,hEjj1 (OK)  
alternative: .83Y7h (OK)  
alternative: 0KjFzu (OK)  
alternative: 3ka59e (OK)  
alternative: 3w.i8q (OK)  
alternative: 6R,IKG (OK)  
alternative: ;,ZkXE (OK)  
alternative: ;awV5M (OK)  
alternative: AwZr1l (OK)  
alternative: CK_lhq (OK)  
alternative: DRXRbZ (OK)  
alternative: H,oAJA (OK)  
alternative: I,.pQX (OK)  
alternative: I0a,PL (OK)  
alternative: LXN1Nr (OK)  
alternative: PFpQ6n (OK)  
alternative: QgopD; (OK)  
alternative: RzuOos (OK)  
alternative: UcrqEX (OK)  
alternative: Zl,pb0 (OK)  
alternative: a5Dvga (OK)  
alternative: bXb8Iy (OK)  
alternative: best!! (OK)  
alternative: cDlUSt (OK)  
alternative: f,CHMJ (OK)  
alternative: lK2Ttc (OK)  
alternative: mKseoz (OK)  
alternative: nViZD2 (OK)  
alternative: ruFvTv (OK)  
alternative: z.2t4B (OK)在这里插入代码片
```

best!!

所以把之前的拼凑起来。

```
flag{you_are_the_best!!}
```




所以这样便把CRC32碰撞的题给做出了咯。

2: GIF图片修复

这是一张GIF

5

题目已经提示这是一张GIF

 flag.zip	2019/3/30 9:32	WinRAR ZIP 压缩...	91 KB
 flag.gif	2019/3/26 19:14	GIF 文件	95 KB
 ._flag.gif	2019/3/26 19:14	GIF 文件	1 KB

猜想一下flag肯定会隐藏在95KB里面（毕竟几个字母都占几KB）

flag.gif
无法打开此文件。

https://blog.csdn.net/qq_43431158

点击图片，但无法打开。

用winhex打开。

U	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
39	61	F4	01	86	01	87	00	00	94	50	63	94	51	63	94	9a 68 20 20 20 20 20 20 20 20 20 20 20 20 20 20
52	63	93	53	64	93	54	65	93	54	64	91	54	64	90	53	52 63 74 85 96 07 18 29 3a 4b 5c 6d 7e 8f 90
64	90	52	63	90	52	63	90	52	62	90	51	62	90	50	63	51 62 73 84 95 06 17 28 39 4a 5b 6c 7d 8e 9f 00
90	50	63	90	50	63	8f	51	64	8f	51	63	8f	51	62	8f	52 63 74 85 96 07 18 29 3a 4b 5c 6d 7e 8f 90
51	62	8f	51	62	8f	51	61	8f	51	61	8f	51	61	90	51	52 63 74 85 96 07 18 29 3a 4b 5c 6d 7e 8f 90
61	91	51	61	90	51	61	90	50	61	8f	50	61	90	50	61	52 63 74 85 96 07 18 29 3a 4b 5c 6d 7e 8f 90
90	50	61	90	50	61	91	50	61	92	50	61	92	4f	62	93	53 64 75 86 97 08 19 2a 3b 4c 5d 6e 7f 80 91
4f	62	94	4f	61	93	4f	60	92	4f	60	90	51	60	90	51	54 65 76 87 98 09 1a 2b 3c 4d 5e 6f 80 91
60	90	52	60	90	52	5f	90	52	5f	90	52	5f	8e	51	60	55 66 77 88 99 0a 1b 2c 3d 4e 5f 70 81
8e	50	61	8e	51	61	8e	51	61	8e	51	62	8e	52	62	8e	56 67 78 89 9a 0b 1c 2d 3e 4f 60 71 82
52	62	8d	53	62	8d	53	62	8c	53	61	8c	53	60	8b	53	57 68 79 8a 9b 0c 1d 2e 3f 50 61 72
61	8a	53	61	89	52	60	8a	51	5f	8c	4f	5f	8d	4f	5f	58 69 7a 8b 9c 0d 1e 2f 40 51 62
8f	4e	5f	8e	4e	5e	8c	4c	5c	8b	4b	5b	8b	4b	58	8c	59 6a 7b 8c 9d 0e 1f 30 41 52
4f	56	90	52	53	93	55	4f	99	5a	55	a1	60	54	ac	65	60 71 82 93 a4 b5 c6 d7 e8 f9
5a	b5	67	5b	bc	65	5c	ce	64	5d	e1	55	63	f6	5a	6d	66 77 88 99 aa bb cc dd ee ff
fc	5f	71	fc	62	74	fd	67	78	fd	6e	7e	fd	74	83	fd	6a 7b 8c 9d ae bf c0 c1 c2 c3 c4
7a	88	fd	7c	8a	fd	82	90	fd	83	91	fd	83	90	fc	7b	6b 7c 8d 9e af b0 b1 b2 b3 b4

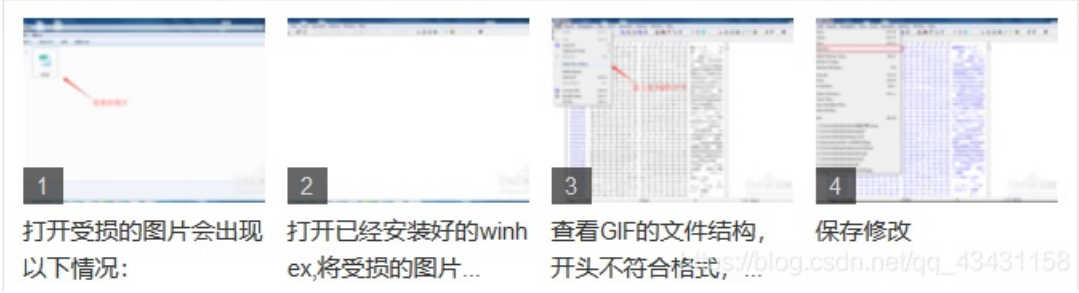
```

7A 00 ED 7C 0A ED 02 30 ED 05 31 ED 05 30 EC 7E 4 y1sy, yj yj u
8D FA 7A 88 F0 75 80 E8 70 7A E3 6C 74 DE 67 6F úz^ðuèèpzãltPgo
DB 64 6C D5 5D 67 CB 56 61 BB 50 4D A5 45 4F 98 ŪdlŌ]gÈVa»PM¥EO~
41 44 8C 42 4C 84 42 4B 7D 40 4C 7B 3D 46 77 37 ADGEBL,,BK}@L{=Fw7
46 76 35 46 74 34 46 73 35 46 73 35 46 72 34 42 Fv5Ft4Fs5Fs5Fr4B
78 31 38 6C 31 3C 64 2E 3B 58 28 33 47 1E 28 34 x1811<d.;X{3G(4

```

题中明明说这是一张GIF，所以应该就是修复GIF图片，查百度、谷歌大法。

如何使用winhex修复受损的图片_百度经验



地址

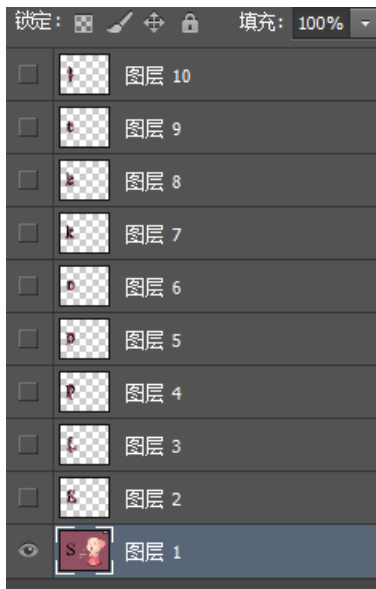
按照上面的操作，并且将文件头改为GIF图片的格式。

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	47	49	46	38	39	61	A2	06	6B	04	F7	FF	00	20	20	20	GIF89a



https://blog.csdn.net/qq_43431158

因为是个动态图，所以下面就得自己想办法让它显示出来。因为我下载了PS软件，所以用这个看会很方便。



欧克，这道题已解开。