

入门知识

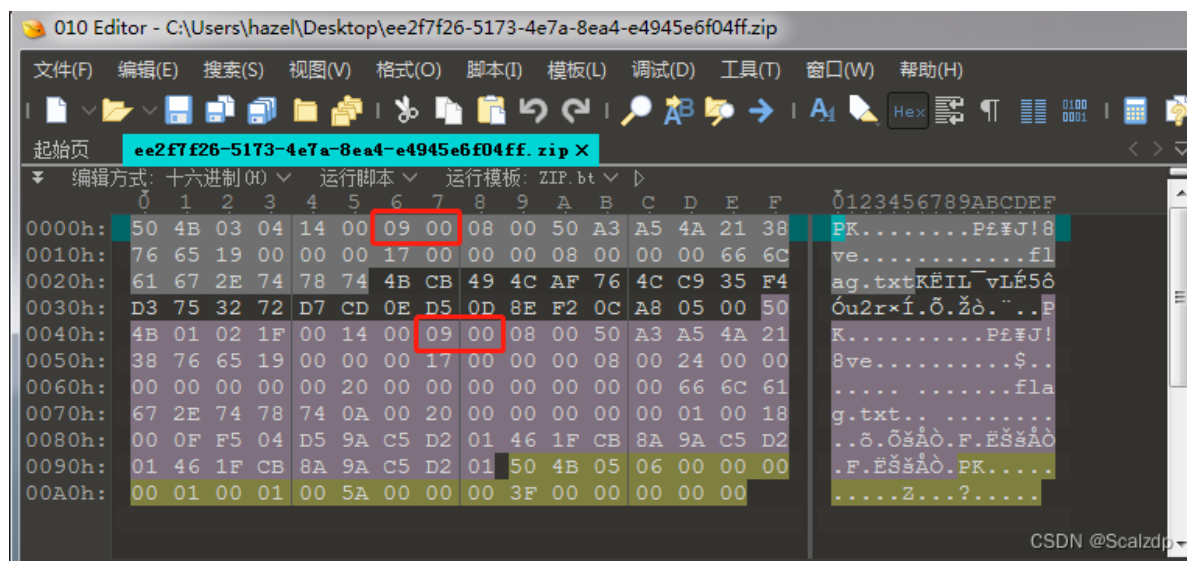
前言

压缩包我们经常接触，用于对文件进行压缩存储/传输。压缩包处理在CTF比赛中是非常重要的一块，因为压缩包中可能包含重要信息：许多CTF题目会将关键信息隐藏在压缩包中，参赛者需要解压并查看其中的内容才能获取有用的线索。解密压缩是常见的CTF技能：参赛者需要掌握各种压缩文件格式的解压方法和工具，以及如何对压缩包进行加密和解密。压缩包处理可以提高解题效率：如果参赛者能够快速解压和查看压缩包中的文件列表和内容，就可以更快地找到关键信息，提高解题效率。对出题方而言，压缩包处理可以增加题目难度：如果一个CTF题目涉及到多个压缩包或复杂的加密算法，那么它会更加具有挑战性，考验参赛者的技术水平和耐心。

平时我们接触到的压缩包，主要有以下类型：zip格式、rar格式、7z格式、tar格式、gzip格式。其中zip和rar是最常见的，其余7z、tar、gzip主要在linux上运行，出题常见于zip和rar，接下来我们简单介绍一下zip和rar两种压缩包。

Zip压缩包

典型特征后缀“.zip”，它的MIME格式为application/zip，zip压缩是一种有损压缩格式，即数据会因压缩而有损失，但是这种损失不会明显影响文件的质量。zip格式可以支持多种压缩算法，如存储、缩小、增强、最佳压缩等，可以根据不同的需求进行选择。zip压缩支持加密压缩。



特点1：数据记录格式：压缩源文件数据区+压缩源文件目录区+压缩源文件目录结束标志

压缩源文件数据库：[文件头+ 文件数据 + 数据描述符]

文件头：50 4B 03 04 :这是文件头标记 (0x04034b50) ，也可看到是“PK...”开头的

压缩源文件目录区为：50 4B 01 02

压缩源文件目录结束标志：50 4B 05 06

需要注意的**加密点**，每组四位数字，只和第二个数字有关系。

第二个数字**为奇数时** ->加密

第二个数字**为偶数时** ->未加密

1. 无加密

压缩源文件数据区的全局加密应当为00 00 (50 4B 03 04两个bytes之后)

且压缩源文件目录区的全局方式位标记应当为00 00 (50 4B 01 02四个bytes之后)

2. 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

3. 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

这种情况可能就只能用暴力破解、明文攻击、CRC32碰撞等方式来破解了。

RAR压缩包

典型特征“.rar”，rar 文件主要由标记块，压缩文件头块，文件头块，结尾块组成。

RAR 文件头 52 61 72 21 1A 07 00

RAR 文件尾 C4 3D 7B 00 40 07 00

伪加密：RAR的伪加密与ZIP的伪加密原理相同，号称伪加密的关键都是一个指定的位标记字段上。

PS：一般RAR伪加密的压缩包用WinRAR打开时都会显示文件头已损坏

在RAR的第24个字节，也就是010 Editor显示的文件结构中的ubyte PASSWORD_ENCRYPTED字段，修改其字段为1即可实现RAR伪加密。

起始页

flag-rar4.rar x

stego.rar

编辑方式: 十六进制(00) v

运行脚本 v

运行模板: RAR.bt v

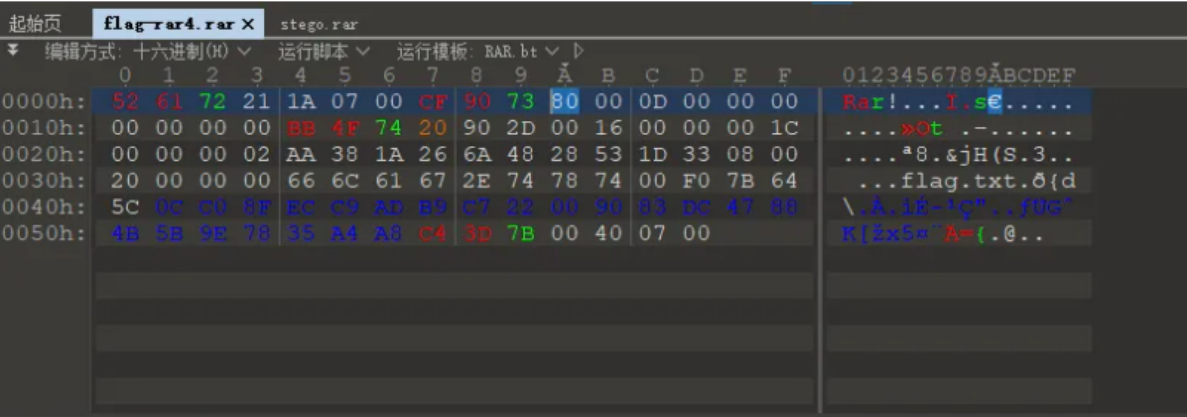
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF	
0000h:	52	61	72	21	1A	07	00	CF	90	73	00	00	0D	00	00	00	Rar!...f.s.....
0010h:	00	00	00	00	BB	4F	74	24	90	2D	00	16	00	00	00	1C	...not\$.-.....
0020h:	00	00	00	02	AA	38	1A	26	6A	48	28	53	1D	33	08	00	...*0.&jH(S.3..
0030h:	20	00	00	00	66	6C	61	67	2E	74	78	74	00	F0	7B	64	...flag.txt.0{d
0040h:	5C	0C	60	8F	EC	89	AD	89	07	22	00	80	83	BC	47	80	\.A.1E~"Q"...f00g"
0050h:	4B	5B	9E	78	35	A4	A8	C4	3D	7B	00	40	07	00			R[2x5e`A=({.0..

模板结果 - RAR.bt

名称	值
struct RarBlock Marker	
struct RarBlock ArchHeader	
struct RarBlock block[0]	
uint16 HEAD_CRC	4FBBh
enum RarBlockType HeadType	FILE_OR_DIR (116)
struct FileHeadFlags HEAD_FLAGS	
ubyte from_PREV_VOLUME : 1	0
ubyte to_NEXT_VOLUME : 1	0
ubyte PASSWORD_ENCRYPTED : 1	1
ubyte FILE_COMMENT_PRESENT : 1	0
ubyte SOLID : 1	0
enum FileDictType DICTIONARY : 3	_128K (1)
ubyte HIGH_SIZE : 1	0
ubyte has_UNICODE_FILENAME : 1	0
ubyte ENCRYPTION_SALT : 1	0
ubyte IS_OLD_FILE_VERSION : 1	0
ubyte EXTENDED_TIME_INFO : 1	1
ubyte _reserved : 1	0
ubyte OLD_VERSION_IGNORE : 1	0
ubyte ADD_SIZE_PRESENT : 1	1
uint16 HeaderSize	45
uint32 RawDataSize	22
struct FileHeadBlock file	
ubyte _reserved[5]	
byte _fcol	

CSDN @Scalzdp

或者修改第11个字节，也就是010 Editor显示的文件结构中的ubyte BLOCK_HEADERS_ENCRYPTED字段的值。修改为1即可造成RAR伪加密。



模板结果 - RAR.bt	
名称	值
struct RarBlock Marker	
struct RarBlock ArchHeader	
uint16 HEAD_CRC	90CFh
enum RarBlockType HeadType	ARCHIVE (115)
struct MainHeadFlags HEAD_FLAGS	
ubyte ARCHIVE_VOLUME : 1	0
ubyte ARCHIVE_COMMENT_PRESENT : 1	0
ubyte ARCHIVE_LOCKED : 1	0
ubyte ARCHIVE_SOLID : 1	0
ubyte NEW_VOLUME_NAMING : 1	0
ubyte AV_PRESENT : 1	0
ubyte RECOVERY_PRESENT : 1	0
ubyte BLOCK_HEADERS_ENCRYPTED : 1	1
ubyte IS_FIRST_VOLUME : 1	0
ubyte _reserved : 5	0
ubyte OLD_VERSION_IGNORE : 1	0
ubyte ADD_SIZE_PRESENT : 1	0
uint16 HeaderSize	13
uint16 _reserved1	0
uint32 _reserved2	0
struct RarBlock block[0]	
struct RarBlock block[1]	

同理解法就是将其对应位置的值修改为0即可实现伪加密rar破解出来。不过一般rar在CTF中出现较少，重点还是zip的压缩包上面做文章。

了解了一些CTF中最常见压缩包格式，接下来我们分享一下如何做CTF题目。其实压缩包的题目，绝大多数是破解密码，其次是不全文件之类的提醒。

常用工具

- 010 Editor
- WinHex
- ARCHPR

实战案例

[如何破解压缩包密码，CTF压缩包处理 rar伪加密-CSDN博客](#)

参考资料

CTFer成长之路-Nu1L战队-Misc部分