MISC-zip压缩包的总结



Qwzf ● 于 2019-11-27 18:57:16 发布 ● 5725 ☆ 收藏 54 分类专栏: CTF 文章标签: CTF 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: <u>https://blog.csdn.net/qg_43625917/article/details/96148661</u> 版权



CTF 同时被3个专栏收录

30 篇文章 6 订阅

订阅专栏



1篇文章0订阅

订阅专栏



4 篇文章 0 订阅 订阅专栏

MISC-zip压缩包的总结

做了那么多的MISC压缩包的CTF题,是时候总结一下经验了。手撕压缩包走起!

1、伪加密

zip中有一位是标记文件是否加密的,如果更改一个未加密zip包的加密标记位,那么在打开压缩包时就会提示该文件是加密的。

压缩源文件数据区: 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 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 6B65792E7478740A0020000000000000010018006558F04A1CC5D001BDEBDD3B1CC5D001BDEBDD3B1CC5D001 压缩源文件目录结束标志: 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

5 7 Offset. 0 2 3 4 6 9 B C D E 1 8 A F Z~"鱂 I 00000000 50 4B 03 04 14 00 01 00 08 00 5A 7E F7 46 16 B5 PK 00000010 80 14 19 00 00 00 17 00 00 00 07 00 00 00 6B 65 . ke 79 2E 74 78 74 OB CE CC 00000020 75 OE 71 AB CE 48 CD C9 y.txt 翁u q沥H蜕 C9 57 28 CE CC 2D C8 49 00000030 AD 28 4D AD 05 00 50 4B 蒞(翁-菼?M? PK Z~IF μ 00000040 01 02 3F 00 14 00 09 00 08 00 5A 7E F7 46 16 B5 2 00000050 80 14 19 00 00 00 17 00 00 00 07 00 24 00 00 00 S . 00000060 00 00 00 00 20 00 00 00 00 00 00 00 6B 65 79 2E kev. 00000070 74 78 74 0A 00 20 00 00 00 00 00 01 00 18 00 65 txt e 08000000 58 F0 4A 1C C5 D0 01 BD EB DD 3B 1C C5 DO 01 BD X餔 判 诫? 判 1 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 Y >

把504B0304后的第3、4个byte改成0000还有 把504B0102后的第5、6个byte改成0000即可破解伪加密。

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

例:

题目下载

Challenge	24 Solves	×
	不成熟的加密	
	10	
奈何本人技术	《差,一伪加密行天下	
Blog.zip		
Кеу		SUBMIT

根据题目,这可能是一道伪加密题。把压缩包文件用winrar打开,看出来b.png进行了伪加密

・此电脑 > 本地磁盘 (D:)	> 网安 > 网安做	城题文件 >	Blog解决			~	U ž	號索"Blo	2
^ 名称	^		修改日期		类型		大小		
🔡 Blog.zip			2019/3/2	3 15:34	WinRA	R ZIP 压缩.		390 KB	
Nog.zip (评估版本	(4						_		×
文件(F) 命令(C) 工具	具(S) 收藏夹(O)	选项(N) 精	^鼎 助(H)						
添加 解压到	 ブラン (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	山 一 一 一	重找	向导	信息) 日描病毒	注释	自解压格	鈨
<u>:</u> ' 名称 [`]	大小	压缩后大小	类型 文件夹		修改时	间	CRC32		
b.png *	207,670	204,652	PNG 文件	;	2019/	3/21 20:	0268F4	47	
🖻 a.png	226,792	193,603	PNG 文件	:	2019/	3/21 21:	E31979	21	
<									>
■ 🖘 已经选择 207,6	670 字节(1 个文件	=)		总计 434,	462 字节	5(2个文件)			.:

把Blog的zip压缩包放进winhex,在最上面没发现伪加密标志位,所以我在最下面发现

1B FF A4 7F 00 50 4B 01 02 3F 00 14 00 00 00 08 ÿ¤ PK ? 00 9A A8 75 4E 21 79 19 E3 43 F4 02 00 E8 75 03 š uN!y ãCô èu S 00 00 00 61 2E 70 6E 67 0A 00 20 00 00 00 00 00 a.png ñë¤≪æßÔ K+r≪ 01 00 18 00 F1 EB A4 AB E6 DF D4 01 4B 2B 72 AB E6 DF D4 01 7F 14 29 AB E6 DF D4 01 50 4B 01 02 æßô)«æßô PK 3F 00 14 00 09 00 08 00 15 A6 75 4E 47 F4 68 02 ? ¦uNGôh 6C 1F 03 00 36 2B 03 00 05 00 24 00 00 00 00 00 1 6+ Ş fô b.png 00 00 20 00 00 00 66 F4 02 00 62 2E 70 6E 67 0A 00 20 00 00 00 00 00 01 00 18 00 62 8D E7 68 E4 b çhä DF D4 01 2F 10 E2 68 E4 DF D4 01 6A 72 8A 8B E1 BÔ / âhäBÔ jrŠ<á DF D4 01 50 4B 05 06 00 00 00 00 02 00 02 00 AE BÔ PK R 00 00 00 F5 13 06 00 00 00 õ

把b.png的504B0102后第五位09改成00,即可破解伪加密,解压即可在b.png中得到flag!!

S STATES						1		C. C
	📲 Blog	解决.zip	(评估版本	5)	_			×
	文件(F)	命令(C)	工具(S)	收藏夹(D) 选项(N) 帮	助(H)	
BLOG	a G							\$ *
b.png	添加 个	解压至	川 测证	t 查	看 删隙	ŧ	查找	ſ
	名称	¥.		大小	压缩后	大小	类型	
BLOG	II						文件夹	
	🖻 b.png	9		207,670	204,	652	PNG文	7件
a.png	📄 a.png	9		226,792	193,	603	PNG文	7件
	<							>
					总计 434	,462 =	字节(2	个文1

2、明文攻击

明文攻击是一种较为高效的攻击手段,大致原理是当不知道一个zip的密码,但是有zip中的一个已知文件(文件大小要大于 12Byte)时,因为同一个zip压缩包里的所有文件都是使用同一个加密密钥来加密的,所以可以用已知文件来找加密密钥,利用 密钥来解锁其他加密文件。

推荐一个工具:APCHPR (可进行爆破/明文/字典/掩码攻击)

例:

题目下载

Challenge	0 Solves		>	\times
		明文 10		
已知明文, 言	言尽于此			
zip				
Кеу			SUBMIT	

📴 zip (评估版本)				- [- ×
文件(F) 命令(C) 工具	具(S) 收藏夹(O)	选项(N) 帮	⁸ 助(H)		
读 添加 解压到	》 	删除	 <u> </u>	 信息 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	注释
\uparrow				·	
	大小	压缩后大小	类型 文件夹	修改时间	CRC32
a blingbling.zip	9,755	9,558	WinRAR ZIP 压缩	. 2019/3/20 21:	4FB0B314
提示.txt	39	39	文本文档	2019/3/20 21:	5C6114DB
ingbling.zip (济	P估版本)			- [
文件(F) 命令(C) 工	具(S) 收藏夹(O)	选项(N)	昏助(H)		
📑 🧰		Î) 🔍	 (i) (i)	E) »
添加 解压到 个	测试 查看	删除	查找 向导	信息 扫描病毒	注释
名称	大小	压缩后大小	类型	修改时间	CRC32
I			文件夹	12 12 12 12	
☐ 提示.txt *	39	51	文本文档	2019/3/20 21:	5C6114DB
🚽 空白文档? ? ?	11,945	9,226	Microsoft Word	. 2019/3/20 20:	F6928DE8
<					>
-			总计 11,984 字	록节(2 个文件)	

在zip文件里有个**未加密的 提示.txt文件**,在blingbling.zip有个**加密的 提示.txt文件**。所以这应该是明文攻击。把未加密的 提示.txt文件压缩成zip压缩包 当作明文。对blingbling.zip进行明文攻击,如下:

🖁 ARCHPR 4.54 Professional Edition – 🗆 🗙							
文件(F) 恢复(R) 帮助(H)							
	☆ 升级 ホキボ型	? 帮助	① 关于	<mark>。</mark> 退出			
D: 1的安1的安顾题又1年121p-bling解决1plin	明又			~			
范围 长度 字典 明文 自动保存 选项 高级 明文选项 明文文件路径: D:\网安\网安做题文件\zip-bling解决\提示.zip 2 ・ 开始于: 0 會 密钥 yrsimmpzrbk- 密钥 zokkhuapolo- 密钥 xloobshxdba- 4							
□ 允许使用二进制文件作为明文 ZIP 档案	尾文件						
状态窗口							
2019/4/5 11:53:11 - ARCHPR 4.54 build 45 2019/4/5 11:53:29 - 文件"D:\网安\网安做题 开。	已启动 频文件\zip-blir	ng解决∖blir	ngbling.zip"E	2打 0			
当前口令: 已用时间: 进度指示器	平均速度: 剩余时间:						
0	%						
ARCHPR version 4.54 (c) 1997-2012 ElcomSoft	Co. Ltd.						

得到**空白文档???**的口令密码:

这个文件的口令	hahaha	ር
十六进制口令	68 61 68 61 68 61	ዓ
这个文件的口令	hahaha	ය
十六进制口令	68 61 68 61 68 61	
平均速度(口令/秒)	2,731,935	
总计口令	84,690 31ms	

打开空白文档???,发现真的是空白的,想吐血。不过查了查百度发现,打开word文件,选择"选项",然后选择"显示"

Word 选项		?	×
常规	✓ 悬停时显示文档工具提示(L)		
显示	始终在屏幕上显示这些格式标记		
校对	□ 制表符(I) →		
保存	□ 空格(<u>S</u>) ····		
版式	✓ 段落标记(M)		
语言	✓ 隐藏文字(D) abc		
轻松访问	□ 可选连字符(Y)		
高级	✓ 对象位置(C)		
自定义功能区	□ 可选分隔符(<u>O</u>) □		
快速访问丁昌栏	□ 显示所有格式标记(<u>A</u>)		
加裁项	打印选项		
信任由心	✓ 打印在 Word 中创建的图形(<u>R</u>) ①		
IDIT.1.0.	□ 打印背景色和图像(<u>B</u>)		
	□ 打印文档属性(P)		
	✓ 打印隐藏文字(Ⅹ)		
	□ 打印前更新域(E)		
	□ 打印前更新链接数据(K)		-
	确完	Π∇	 消

在隐藏文字和打印隐藏文件前打对勾,然后确定,然后打开空白文档???,就能看到flag了!!

3、crc32碰撞

CRC32:CRC本身是"冗余校验码"的意思,CRC32则表示会产生一个32bit(8位十六进制数)的校验值。****

在产生CRC32时,源数据块的每一位都参与了运算,因此即使数据块中只有一位发生改变也会得到不同的CRC32值,利用这个 原理我们可以直接*爆破出加密文件的内容*

例:

题目下载



主要就是crc, 贼简单, 但别死脑筋



Key

SUBMIT

 \times

根据题目可知,这道题可能是crc32碰撞。下载该压缩包,用winrar打开

🌠 crc32.zip (评估版	(本)				– 🗆 X
文件(F) 命令(C) 工	具(S) 收藏夹(O)	选项(N) 幕	§助(H)		
<mark>神神</mark> () 添加 解压到 测 ↑		除 查找	向导信息	日描病毒 注释 自解压格式	
名称	大小	压缩后大小	类型	修改时间 CRC32	
II			文件夹		
📄 flag.txt *	102	114	文本文档	2019/3/19 20: 7F0170	62D
3.txt *	6	18	文本文档	2019/3/19 20: 4FA83	D8C
2.txt *	6	18	文本文档	2019/3/19 20: 127F1	984
1.txt *	6	18	文本文档	2019/3/19 20: 7F616	EE3

会发现4个加密的txt文件,有三个大小为6的,一个存放flag大小为102的。**crc32碰撞,碰撞文件的大小一般不大于6,大于6的 一般碰撞不出来。且位于同一压缩包的文件,文件密码相同。**所以对1.txt 2.txt 3.txt进行crc32碰撞

还有一款很好用的6位的CRC32爆破

附上神器: https://github.com/theonlypwner/crc32 具体使用方法:

python crc32.py reverse 你的crc32密文

密文记得加上0x变成16进制,三个txt文件碰撞结果如下

\$ python crc	32.py re	everse 0x7f616e	ee3
4 bytes: {0x	fc, Öxf3	3, 0x48, 0x10	
verification	checksı	um: 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:	161PcW	(OK)	
alternative:	m6paxN	(OK)	
alternative:	pymQwW	(OK)	
alternative:	xo4nzk	(OK)	
alternative:	you_ar	(OK)	

2.txt

<pre>\$ python crc3</pre>	32.py re	everse 0x127f1984	
4 bytes: $\{0x($	Oc, Oxa9	0, 0xe2, 0xfd}	
verification	checksu	ım: 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)	

\$ python crc3	32.py re	everse 0x4fa83	d8c
4 bytes: {0x7	7e, Oxfa	n, 0xeb, 0x0a}	
verification	checksu	um: 0x4fa83d8c	(OK)
alternative:	0KjFzu	(OK)	
alternative:	3ka59e	(OK)	
alternative:	AwZr11	(OK)	
alternative:	CK_1hq	(OK)	
alternative:	DRXRBZ	(OK)	
alternative:	LXN1Nr	(OK)	
alternative:	PFpQ6n	(OK)	
alternative:	Rzu0os	(OK)	
alternative:	UcrqEX	(OK)	
alternative:	a5Dvga	(OK)	
alternative:	bXb8Iy	(OK)	
alternative:	cDlUSt	(OK)	
alternative:	1K2Ttc	(OK)	
alternative:	mKseoz	(OK)	
alternative:	nViZD2	(OK)	
alternative:	ruFvTv	(OK)	

在碰撞的内容中,找有意义的字符。1.txt中**"you_ar**" 2.txt中"**e_the_**" 3.txt中未发现有意义的字符。做到这一步,再次想吐血。还 好我用notepad++打开碰撞脚本,发现一组特殊之处

📔 D:\	\网安\网安工具\脚本\crc脚本\crc32\crc32.py - Notepad++	_	×
文件(F)) 编辑(E) 搜索(S) 视图(V) 编码(N) 语言(L) 设置(T) 工具(O) 宏(M) 运行(R) 插件(P) 窗囗(W) ?		х
🕞 🚽) 🔚 🛍 📑 🐚 🚔 🔏 👘 🛅 🗩 🗲 📾 🍢 🔍 🔍 🖼 🖼 💷 11 🏋 🐺 💹 🖉 💷 👁 💌 🗩 🖼 🖷		
= 456	6. txt🗙 🔚 crc32. py🛛 🔚 6. py🗵		
4	<pre>#usage: python crc32.py reverse 0xffffffff (the crc)</pre>		^
5			
6	import argparse		
	import os		
8	import sys		
10	Dermitted characters = set (
11	map(ord, 'abcdefghiiklmnopgrstuvwxvzABCDEFGHIJKLMNOPORSTUVWXYZ01234567890 '))	# \w	
12			
13	testing = False		
14	-		
15	args = None		
16			
17			
18	Fight get_poly():		
19	poly = parse_dword(args.poly)		
20	if args.msb:		
21	<pre>poly = reverseBits(poly) chaple2(rely)</pre>		
22	check32 (poly)		
23	- recurn pory		
25			
2.6	Edef get input():		
27	if args.instr:		
28	return tuple(map(ord, args.instr))		
29	with args infile as f # pragma: no cover		~
<			 >
Python	file length : 12,580 lines : 375 Ln : 11 Col : 79 Sel : 0 0 Windows (CR LF) UTF	-8	INS

想着是不是因为脚本里缺少特殊字符,而3.txt里有特殊字符。所以碰撞不出3.txt的内容。加上特殊字符后,碰撞结果如下

\$ python crc32.py re	verse 0x4fa83d8c
4 bytes: {0x7e, 0xfa	, 0xeb, 0x0a}
verification checksu	m: 0x4fa83d8c (OK)
alternative: (1XkkR	(OK)
alternative: ,hEjjl	(OK)
alternative: .83Y7h	(OK)
alternative: /8rh,q	(OK)
alternative: /u_UAy	(OK)
alternative: OKjFzu	(OK)
alternative: 3ka59e	(OK)
alternative: 3w.i8q	(OK)
alternative: 6R,IKG	(OK)
alternative: ;,ZkXE	(OK)
alternative: ;awV5M	(OK)
alternative: ?ejW4.	(OK)
alternative: AwZrll	(OK)
alternative: CK_lhq	(OK)
alternative: DRXRBZ	(OK)
alternative: ENV?XW	(OK)
alternative: H, oAJA	(OK)
alternative: I,.pQX	(OK)
alternative: IOa,PL	(OK)
alternative: LXN1Nr	(OK)
alternative: M(3qP;	(OK)
alternative: N5)N{s	(OK)
alternative: PFpQ6n	(OK)
alternative: QgopD;	(OK)
alternative: RzuOos	(OK)
alternative: UL(P	(OK)
alternative: UcrqEX	(OK)
alternative: XP)n;R	(OK)
alternative: Zl, pb0	(0K)
alternative: abDvga	(0K)
alternative: bXb8ly	(0K)
alternative: best!!	(UK) (OV)
alternative: cDIUSt	(0h)
alternative: f, CHMJ	(UK) (OV)
alternative: 1}!)!V	(0h)
alternative: IK2Itc	(0h)
alternative: mKseoz	(0K)
alternative: nv12D2	(0K)
alternative: owv(bg	(0K)
alternative: ruFvlv	(0K)
alternative: 2.214B	(0K)
arternative. 225 (3V	

找到3.txt中的有意义的字符了,好开心! ! ! 3.txt中的有意义字符"**best!!**",结合1.txt的"**you_ar**" 2.txt的"**e_the_**"得到flag.txt的密码: "**you_are_the_best!!**",输入密码打开后发现

 flag.txt - 记事本

 ý件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

 你在找这个吗?好像被加密了呢,先解密一下吧。

 ZmxhZyU3QndIMWNvbWUIMjB0byUyMHNoYWxvdSUyMGFucXVhbiU3RA==



看起来解密结果符合url编码,所以url解码,得到最终flag如下:

Native/Unicode Native/UTF-8	Native/ASCII	URL转码		
Url:				编码结果:
flag{we1come to shalou anquan}			 encodeURI encodeURIComponer URL编码 > URL编码 	flag% <u>7Bwe1come%20to%20shalou%20anquan%7D</u>

4、爆破/字典/掩码攻击

把这三种归位一类是因为这三种方法在本质上都是逐个尝试,只不过待选密码的集合不同

爆破:顾名思义,逐个尝试选定集合中可以组成的所有密码,知道遇到正确密码

字典:字典攻击的效率比爆破稍高,因为字典中存储了常用的密码,因此就避免了爆破时把时间浪费在脸滚键盘类的密码 上

掩码攻击:如果已知密码的某几位,如已知6位密码的第3位是a,那么可以构造??a???进行掩码攻击,掩码攻击的原理相当于构造了第3位为a的字典,因此掩码攻击的效率也比爆破高出不少

例:

题目下载



所以应该要暴力破解,把下载的压缩包放进ARCHPR(爆破工具)里,选择字典攻击(字典攻击比暴力攻击用时少),

得出文件密码口令

🔏 ARCHPR 4.54 - 82%	🔏 ARCHPR 4.54 - 82% —										
文件(F) 恢复(R) 帮助(H)											
診 · ぷ パ パ 打开 开始: 停止 想	💰 🥸 基准测试 升级	? 帮助 :	 ① 关于 	退出							
加密的 ZIP/RAR/ACE/ARJ 文件 D:\网安\网安做题文件\flag2解决\fla		~									
范围 长度 字典 明文 自动作 字典选项 字典文件路径:	范围 长度 字典 明文 自动保存 选项 高级 字典选项 字典文件路径:										
D:\网女\网女工具\吸解攻击工具\E 口令已成功恢复! Advanced Archive Pa	ssword Recovery 统计信	• 开始打ち 息:	×	0							
总计口令	199,974		-								
	46ms										
平均速度(口令/秒)	4,347,260			^							
这个文件的口令	summer		G								
十六进制口令	73 75 6d 6d 65 72		<u>a</u>	~							
┏ 保存											
	82%										
ADOUDD mention A EA (a) 1007 0010 Floore	Soft Co. 1td										

输入密码,解压文件,得到一个文件夹。然后文件夹里有好几个文件夹,所以开启找可疑之处的历程

最终,我发现有个地方比较可疑

```
function checknum(num) {
    if(num.length==""")
    {
        document.getElementById("usernum").innerHTML="<font color = 'red'>请输入账号</font>"
        myform.num.focus();<!--6666c61677b73756d6d6572217d-->
        return false;
    }
    else{
```

看着比较像Base16加密,所以进行Base16解密,得到flag

Base16编码解码



flag{summer!}

5、杂类

例1:

题目下载

Challenge	0 Solves			>
		简申zip		
		15		
提交格式SL]豕恨讲宄规定 }			
成本的 Line zip.zip]永仅讲究规定 }			
現代に、18 提交格式SL zip.zip]永(皮)开充决反E }			
レ 提交格式SL zip.zip Key]永(皮)开究状况E }		SUBMIT	

很明显,这个题并没有给出有效提示(只给了提交格式)。下载后,用winrar打开发现一个zip.txt文件,打开后

根据504B0304可以看出这应该是一个压缩包的16进制编码,然后查询一下压缩包16进制编码格式,把不符合16进制编码的改成 对应的16进制编码。改完后,把16进制编码粘贴在winhex里进行对压缩包的恢复

注意选择16进制粘贴格式

UTF-16 Unicode		
ANSI-ASCII ASCII Hex		
🖪 确定(<u>O</u>)	取消(<u>A</u>)	<i>《</i> 帮助(<u>H</u>)

🧱 WinHex - [未命名]

选择剪贴板格式

🧱 文件(F) 编辑(E) 搜索(S) 导航(N) 查看(V) 工具(T) 专业工具(I) 选项(O) 窗口(W) 帮助(H)

 \times

案件数据	🗋 🗛 🖪 🖉 🖉	🔌 😭 📑) 🖣 🛱	ß	101- 010	2	A M	HEX	¢₽ (HEX	-	-100	+ -		l 🕹 🕹 🧼 🥅 🔎	🦚 💒 🥨
文件(L) 编辑(D)	未命名																
	Offset	0 1	2	3 4	5	6	7	8	9	10	11	12	13	14	15	ANSI	ASCII
	00000000	50 4B	03	04 14	00	00	08	08	00	57	9A	74	4E	08	F6	PK V	NštN ö
	00000016	F8 5D	12	00 00	00	10	00	00	00	80	00	00	00	66	6C	ø]	fl
	00000032	61 67	2E	74 78	74	AB	CA	48	CC	4B	AF	2C	AD	C8	4C	ag.txt«ÊHÌ	Х [−] ,-ÈL
	00000048	CC 2F	4 F	CC AB	CA	04	00	50	4B	01	02	3F	00	14	00	Ì/OÌ«Ê PK	?
	00000064	01 08	08	00 57	9A	74	4E	08	00	F8	5D	12	00	00	00	WštN 🕫	ø]
	00000080	10 00	00	00 08	00	24	00	00	00	00	00	00	00	20	00	Ş	
	00000096	00 00	00	00 00	00	66	6C	61	67	2E	74	78	74	0A	00	flag	.txt
	00000112	20 00	00	00 00	00	01	00	18	00	F3	6F	51	AE	0E	\mathbf{DF}	ć	်ဝQ© ပြီ
	00000128	D4 01	F3	6F 51	AE	0E	\mathbf{DF}	D4	01	AB	19	BA	19	0C	\mathbf{DF}	Ô Ó O QE BÔ <	к° В
	00000144	D4 01	50	4B 05	06	00	00	00	00	01	00	01	00	5A	00	Ô PK	Z
	00000160	00 00	38	00 00	00	00	00									8	

然后保存,用winrar打开恢复后的压缩包,结果发现一个flag.txt文件竟然是加密的,再次想吐血!

🌠 123 (评估版本)			_	
文件(F) 命令(C) I	[具(S) 收藏夹(O) 选项(N)	帮助(H)		
☆ か 加 か か		・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・) (三) 人 病毒 注释 自解	玉格式
名称	大小 压缩后大	小 类型	修改时间	CRC32
Ш.,		文件夹		
📄 flag.txt *	16	18 文本文档	2019/3/20 19:	5DF80008

🇱 WinHex - [123]								- 0	×
🇱 文件(F) 编辑(E)	搜索(S) 导航(N) 查看(V)	工具(T) 专	业工具(I) 🖯	选项(O) 窿	窗口(W) 帮助(H)	19.6 x64 _	e x
案件数据	۵ 😂 🗔 🤜 🗅 🗍	ا 📔 🖆 📚	- h e c	B 010	🎢 🚧 💏 '	🍰 🏰 📔 → -	H 🔶 🔶	🕹 촂 🖿 🔎 🎼 💕	
文件(<u>L)</u> 编辑(<u>D</u>)	123								
	Offset	0 1 2	3 4 5	567	89	10 11 12 1	13 14 15	ANSI ASCI	123
	00000000	50 4B 03	04 14 00	0 00 08	08 00	57 9A 74 4	4E 08 F6 PI	K WštN	C:\Us
	00000016	F8 5D 12	00 00 00	0 10 00	00 00	08 00 00 0	00 66 6C ø]f	
	00000032	61 67 2E	74 78 74	4 AB CA	48 CC	4B AF 2C A	AD C8 4C a	g.txt«EHIK ,-E	168/4
	00000048		00 57 91	A 04 00	08 00	F8 5D 12 (Wétn al	字节
	00000080	10 00 00	00 08 00	0 24 00	00 00	00 00 00 00	00 20 00	S	缺省维
	00000096	00 00 00	00 00 00	0 66 6C	61 67	2E 74 78 7	74 0A 00	flag.txt	司城的:
	00000112	20 00 00	00 00 00	0 01 00	18 00	F3 6F 51 A	AE OE DF	- óoQ®	1 +***
	00000128	D4 01 F3	6F 51 AB	E OE DF	D4 01	AB 19 BA 1	19 OC DF Ô	° » Ôũ 300ò	「」
	00000144	D4 01 50	4B 05 00	6 00 00	00 00	01 00 01 0	00 5A 00 Ô	PK Z	
	00000160	00 00 38	00 00 00	0 00 00				8	04创建日
									47:58
) 4 क्राह ्य
									47:06
									唐]伍:
									ISCITATA.
									∖獲識:
									c偏移均
									=毎页=
									当前留
									窗口息
	<							>	剪購載
	页1/1		偏移地址:	64		= 1 选块:		无 大小:	无

把01改成00后保存,再次用winrar打开,发现flag.txt变成了未加密,开心!找到flag了!!

123	(评估版本	🗐 flag	.txt - 记事	砵	_		×
文件(F)	命令(C)	文件(F)	编辑(E)	格式(O)	查看(V)	帮助(H)	
添加	解下到	zhang	yuxiao	wanzi			^
<u>та-зн</u>	1917-22-3						
名称	~						
1.							
📄 flag.t	xt						
							\checkmark

例2:

题目下载



这个题给了提示,一个公式f(x)=(x-n)%26+97,看着有点像凯撒公式。下载后,用winrar打开压缩包,进入"皮"文件夹发现两个txt 文件

[外链图片转存失败(img-BfScdxNr-1563265524323)(http://i1.fuimg.com/690374/5a1c93aa0a869757.png)]

打开两个txt文件

🥘 key.t	xt - 记事	本	_			🥘 提示	.txt - 记号	本	_		\times	
文件(F)	编辑(E)	格式(O)	查看(V)	帮助(H)		文件(F)	编辑(E)	格式(O)	查看(V) 帮助(H)	
cipkey1 cipkey2 cipkey3	I: yrsir 2: zokł 3: xloo	nmpzr ‹huapo ›bshxdł	bkcoko iloaeku oamnq	d n	^	这么家 : i	伙不小	心留下	了小弟	的名字	2	^
					\sim							\sim

key.txt应该就是凯撒加密的密文了,而提示.txt可能是密钥。然后,我开始理解题目提示的凯撒公式**f(x)=(x-n)%26+97**,我理解这 是个加密公式,*f(x)是密钥,x是明文,n是密文*。然后我写出对应的解密公式**str=((c1-97)+(key-97))%26+97**,***str是明 文,c1是密文,key是密钥。***由于写这篇blog时,我还不会写python脚本,于是我写了个c语言的

```
#include <stdio.h>
#include <stdlib.h>
int main()
   char key[20];
   char c1[20]="yrsimmpzrbkcokd",c2[20]="zokkhuapoloaeku",c3[20]="xloobshxdbamnqn",str1[20],str2[20],str3[20];
   for(j=1;;j++)
       printf("请输入密钥:");scanf("%s",key);
       printf("密文1:%s\n密文2:%s\n密文3:%s\n",c1,c2,c3);
       //printf("密文1:%s",c1);//scanf("%s",c1);
      // printf("密文3:%s",c3);//scanf("%s",c3);
       printf("明文1:");
       for(int i=0;i<strlen(key);i++)</pre>
           str1[i]=(c1[i]-97+key[i]-97)%26+97;
           printf("%c",str1[i]);
       printf("\n");
       printf("明文2:");
       for(int i=0;i<strlen(key);i++)</pre>
           str2[i]=(c2[i]-97+key[i]-97)%26+97;
           printf("%c",str2[i]);
       printf("\n");
       printf("明文3:");
       for(int i=0;i<strlen(key);i++)</pre>
           str3[i]=(c3[i]-97+key[i]-97)%26+97;
           printf("%c",str3[i]);
       printf("a b c d e f g h i j k l m n o p q r s t u v w x y z");
       printf("\n");
       printf("1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26");
       printf("\n");
       printf("26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1");
       printf("\n\n");
```

输入第一个密钥,运行结果如下

■ D:\网安\网安做题文件\zip2凯撒解决\c语言代码\bin\Debug\123.exe

青输入密钥:i														
密文1:yrsimmpzrbkcokd														
密文2:zokkhuapoloaeku														
密文3:xloobshxdbamnqn														
明文1:g														
明文2:h														
明文3:f														
a b c d e f g h	i j k l m	nopqr	stuv w x y z											
$1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8$	9 10 11 12 13	14 15 16 17 18	19 20 21 22 23 24 25 26											
26 25 24 23 22 21 20 19	18 17 16 15 14	13 12 11 10 9	8 7 6 5 4 3 2 1											
请输入密钥:_														

然后想到明文3前四个字母应该就是f、l、a、g,而密文3前四个字母是x、l、o、o。所以第二个密钥应该是f(x)=(l-l)%26+97=a;

第三个密钥应该是f(x)=(a-o)%26+97=m;第四个密钥应该是f(x)=(g-o)%26+97=s;

请密密密明明明	输入 文2 文3 文2 文3 文2 文3	、密 :yrs :zol :xlo :gre :how :fla	钥:: simr kkhu bobs ea wc ag	iams npzi lapo shxo	s rbko oloa dbar	coko aeku Ingr	1 1 1																		
a	b	с	d	e	f	g	h	i	j	k	1	m	n	o	р	q	r	s	t	u	v	w	x	y	z
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

然后明文1第五个字母应该是t, 密文1第五个字母是m。第五个密钥应该是f(x)=(t-m)%26+97=g+1=h;

注意:

明文-密文=正值,从前往后数,密钥=正值(即英文字母序号)+1;

明文-密文=负值,从后往前数,密钥=负值的绝对值(即26-英文字母序号);

找到密钥序号对应的字母

明文-密文=零,密钥=a;

请密密密明明明	输入1 文2 文3 文2 文3	密 yrs zol xlo gre how fla	钥:i simm khu pobs eat wco agi	iams npzi iapo shxo	sh cbko oloa lbar	coko aeku nngr	1 1 1																		
a	b	c	d	e	f	g	h	i	j	k	1	m	n	o	р	q	r	s	t	u	v	w	x	у	z
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

请密密密明明明	諭入1 文22 文3 文2 文2 文3	密 yrs zol xlc gre how fla	钥:i simm skhu bobs eatm vcou agis	iams npzr napc shxc nanl 11di ssl	shal obko loa lban like isoo xiao	louh coko aeku anqr eyou cool cxue		ker																	
a	b	c	d	e	f	g	h	i	j	k	1	m	n	o	р	q	r	s	t	u	v	w	x	y	z
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

下面也有个别人写好得python脚本,果然写python脚本才更简洁、更容易。我要努力学python了!!



以上便是我做CTF压缩包类型题的总结,总结许多收获也许多。小白进阶ing!!!