## CTF杂项知识点

# 原创



CTF 专栏收录该内容

1 篇文章 0 订阅 订阅专栏

CTF-杂项

文件操作与隐写

1.File命令

当文件没有后缀名或者有后缀名而无法正常打开时,根据识别出的文件类型来修改后缀名即可正常打开文件

使用场景:不知道后缀名,无法打开文件

root@kali:~# file ctf ctf: PNG image data, 731 x 672 //png文件

#### 2.winhex

通过winhex程序中可以查看文件头类型,根据文件头类型判断出文件类型

使用场景: windows下通过文件头信息判断文件类型

常见的文件头类型如图所示

文件类型。	文件头。
JPEG (jpg)₽	FFD8FFE1.
PNG (png).	89504E47+
GIF (gif)₀	47494638.
TIFF (tif).	49492A00+
Windows Bitmap (bmp).	424DC001+
ZIP Archive $(zip)_{v}$	504B0304.
RAR Archive (rar).	52617221.
Adobe Photoshop (psd).	38425053.
Rich Text Format (rtf).	7B5C727466~
XML (xml).	3C3F786D6C+2
HTML (html).	68746D6C3E*
Adobe Acrobat (pdf).	255044462D312E+
Wave (wav)⇔	57415645+
pcap (pcap) + https://	64D3C2B1A@in_46555037

#### 3.文件头缺失/错误

通常文件无法正常打开有两种情况,一种是文件头缺失,一种是文件头错误。可以使用winhex打开文件,然后修改正确。



当file一个文件的时候显示data就是该文件头缺失或错误。

#### 4.Binwalk文件分离工具

Binwalk是Linux下用来分析和分离文件的工具,可以快速分辨文件是否由多个文件合并而成,并将文件进行分离。如果分离成功 会在目标文件的目录。<mark>有压缩包自动解压</mark>

同目录下生成一个形如\_文件\_extracted的文件目录,目录中有分离后的文件。

用法:

#### 分离文件: binwalk -e filename

root@kali2:~/o	tf# binwalk ans.	jpg		-0
DECIMAL	HEXADECIMAL	DESCRIPTION		
0 8232 19610	0×0 0×2028 0×4C9A	JPEG image data, JFIF TIFF image data, big- Copyright string: " (	standard 1.01 endian c) 1998 Hewlett-ഏകുൾകൂടംഫെ6ുണ്ണ	aanyee067

从0x0到0x2028(十六进制数)是一个JPG图片,从0x2028到0x4C9A是一个TIF文件。

#### 5.foremost文件分离

如果binwalk无法正确分离文件,可以使用foremost,将目标文件复制到kali中,成功执行后,会在目标文件的文件目录下生成我 们设置的目录,目录中会按文件类型分离出文件。有压缩包不会自动解压

#### 用法:

foremost 文件名 -o 输出目录名

# root@kali2: ~/ctf# foremost oddpic.jpg - 0 oddpic Processing: oddpic.jpg |\*|

#### 6.dd文件分离

当文件自动分离出错或者因为其他原因无法自动分离时,可以使用dd实现文件手动分析。

使用情况:



题目将jpg文件和zip文件分成两部分打乱排序。这时候使用binwalk和foremost是分离不出来的。

格式:

dd if=源文件 of=目标文件名 bs=1 skip=开始分离的字节

参数说明:

if=file #输入文件名,缺省为标准输入。

of=file #输出文件名,缺省为标准输出。

bs=bytes #同时设置读写块的大小为bytes,可代替ibs和obs。

skip=blocks #从输入文件开头跳过blocks个块后再开始复制。



命令一:

# dd if=1.txt of=2.txt bs=5 count=1

结果一:

2.txt:
12345

命令二:

dd if=1.txt of=3.txt bs=5 count=2

结果二:

3.txt: 1234567890

命令三:

dd if=1.txt of=4.txt bs=5 count=3 skip=1

结果三:

4.txt:	
67890abcdefg	

一张sim.jpg已经通过binwalk分析出0-22895是jpg, 22895-23046是zip使用dd将其分离 dd if=sim.jpg of=111111.zip bs=1 count=23046 skip=22895

#### 7.文件合并

(1) Linux下的文件合并

使用场景:出题人把一张图片分开好几个碎片给你。或对文件名相似的文件要进行批量合并。

格式: cat 合并的文件 > 输出的文件

root@kali2: ~/ctf/cat# cat chapter01 chapter02 chapter03 > book
root@kali2: ~/ctf/cat# cat chapter\* > book1

完整性检测:Linux下计算文件md5(文件的md5题目会告诉你的):

md5sum 文件名

reborn@Oooo:/mnt/d/forkali\$ md5sum sim.jpg d09e8a07b6dedb0633aa3c432f931362 sim.jpg

(2) Windows下的文件合并

使用场景:出题人把一张图片分开好几个碎片给你。或对文件名相似的文件要进行批量合并。

格式: copy /B 合并的文件名 输出的文件名

D:\CTF\copy>copy	/B chapter01+chapter02+c	hapter03 book
chapter01		
chapter02		
chapter03		
已复制 1	个文件。	
D:\CTF\copy>copy chapter01 chapter02	/B chapter× book1	
chapter03 已复制     1	个文件。 <sup>https://blog.csdn.ne</sup>	t/weixin_46555037

完整性检测: Windows下计算文件md5(文件的md5题目会告诉你的):

certutil -hashfile 文件名 md5

reborn@0000 D:\forkali\tmpp\15 # certutil -hashfile 2.gif md5 MD5 的 2.gif 哈希: 799b44c761aec7aa1afa5499ac8e3e6b CertUtil: -hashfile 命令成功完成。

图片隐写

**1.Firework** 

用十六进制编辑器打开文件时会看到文件头部中包含firework的标识,通过firework可以找到隐藏图片。firework可以用来查看gif 动图的每一帧图片,或者一张图的每一个图层。例如flag,密码等等。

#### 使用场景: 查看隐写的图片文件

	media Fineworks 8 - (547b5ad111244.png @ 100% (wett)filag_woldy_s _e)") 網驗(E) 税量(V) 認局(S) 得設(M) 文本(T) 命令(C) 論願(T) 聖目(W) 解約(F)	
	Stoldstalden:	
	文本     ····································	Bolt Hatt     Hatt     Hatt     Solution
O Macron 文中(F) 多	nedia Fireworks 8 - [54765ad131244.png @ 200%] 発電(5)	001
25 1 1 1 1 1 1 1 1 1 1 1 1	54755ad131244.prg*	
QP.		
	wctf(flag woldy s. email)	a writing.ed.
	wctf(flag_woldy_s _email) 取消不相	➡ ■ ● ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■
	wctf{flag_woldy_s _email} 取消不相	►的图层
	wctf{flag_woldy_s _email} 取消不相	► 的图层
	wctf{flag_woldy_s _email} 取消不相	
	wctf(flag_woldy_s_email) 取消不相	
	wctf(flag_woldy_s_email) 取消不相	

#### 2.Exif

Exif按照JPEG的规格在JPEG中插入一些图像/数字相机的信息数据以及缩略图像.可以通过与JPEG兼容的互联网浏览器/图片浏 览器/图像处理等一些软件来查看Exif格式的图像文件.就跟浏览器通常的JPEG图像文件一样.

图片右键属性,查看Exif或者查看详细信息,在相关选项卡中查找flag信息。

属性	值	-
说明		
标题		-
主题		
方線	65 65 65 65 65	
香油	fleg{this is exif fleg}	
来源		
作者	abc	
拍攝日期		
程序名称	ACD Systems 数码成像	
获取日期		
版权		
图像		
图像 ID		
尺寸	659 x 460	
宽度	659 像素	-
rater eater	100 44 14	

规	安全	详细信息 以前的版本			
属性	ŧ	值	^		
说	明				
标题	5	password is ChApter3			
主题		password is ChApter3			
分级	ž	****			
标记	3				
备注	E				
来	源	5			
作書	Ť				
拍摄	拍摄日期 2011/11/3 11:59				
程序	洺称	Adobe Photoshop CS5 (12.0x20100115			
获闻	口期				
版权	Z				
2	(a)				
图像	t ID				
分别	率	500 x 408			
宽厚	Ξ	500 像素			
高度	Ē	408 像素			
水平	分辨率	300 dpi			
垂直	分辨率	300 dpi			
位法	酒店	24	¥		
	属性和个人	信息			

#### 3.Stegsolve

当两张jpg图片外观、大小、像素都基本相同时,可以考虑进行结合分析,即将两个文件的像素RGB值进行XOR、ADD、SUB等操作,看能否得到有用的信息,StegSolve可以方便的进行这些操作。

使用场景:题目给出两张基本一样的图片

1.打开一张图片,点击analyse->Image combiner



2.在弹出的窗口中点击左右按钮选择处理方式,点击save保存有价值的结果。



3.给出两张一样的图片的时候可以考虑使用StegSolve来将两张图片进行一个相加相减异或的操作。

例题:

题目给出一张图片



使用十六进制打开这张图片,发现一个下载链接

D:\forka	ali\tmpp\18\女神\1.png - Notepad++ [Administrator] - 🗗 🗙
文件(E) 编	編集(E) 接索(S) 视图(M) 通言(L) 设置(D) 工具(Q) 宏(M) 运行(B) 插件(P) 窗口(W) 2 X
	[1] [2] [3] [4] [4] [4] [4] [4] [4] [4] [4] [4] [4
🔚 new 1 🔀	📄 bak trtl 🔚 filajp; jpgl 🛃 hexdata trtl 🔚 gif012 🔚 1. pagl
	b曽XIESC依弜; ^
	c?乙"鬯询娟獦E0kx寡ENOX傟STT玭]?岿 ?裕鄯o_傀6g、MAK# f ?ESCg惍"DC1d1? ^ 腢坍:
	閟专珞y <p_翁ack k"y?媯昻f窄諟b俙襥belistxn1??拲[null舓??<="" td=""></p_翁ack>
714	T4玢?象4NULSUB"L暲宀4??: 蘥灖缩嵬閯6e鹝1酝\咚?? 些 黭寋扥墨劾ENO 颂鱛qERS独?Q-
	/イirG?颕S N?DCBd VT?a4爭F/
715	殽奊
716	?坠錟DLESOH卓ETB 轄謿DJ)RS?淫?鈫XZ犂DC24眩淟 鹓 襀z?)#DCBEr灕STw齄犷闱?
	慧絑k 庀胆NULPCAN樸窈嫂DC4EOTc? <sub>T</sub> I癉<彖ESC核艋b絟SYN%瘁袉)禐,?2DLE圝rSIY鉖SOH
	&DC1? 园碗餢哢1EM絳採嚈?顄VT觲风?DC13紡裓?SO襰-葡輦傽CAN? C -ō? 亪鏧魲钪 抹
	突Μ熤u
	桾ŊUL?n埥m逽洽D檞?瞙.BSk瑼ŊULU剳椟o ? ŊUL\疻ETB4?*UhC+v ??BS\$嶔b?}ESo
	NAK~ 牧埽#筋G盡o淜,V q息?M锢BSg皂F? 橢DLE7 芋> 濖< :- 橋1 嗎 剧 艤郪+ 埤?k ENO 物 痻?? (
	NULQM/U與田陊Q貳缂林]"甜 t薄ENO@ASYN菀BELd?IrS椘湚ETX舐EF)%?铨RSu槼蹈欏PJ 睗
	錗GS调I鐠i/A槢矓R煡t?]OCAN?: : 嶟 <ds???头痛,gw  td="" 挦!褯??{c譥<="" 蝴饤奎孱圓撥譛~dc3cnul=""></ds???头痛,gw >
	頭識J亚?趦u澗k驲DCLEM?v嘿ZkSTXvwGSeKNUL(訣 d"\p <rs]5闖?b 3它="" ?歝+槽="" td="" {梘鱔?<=""></rs]5闖?b>
	STX 握  聚%备 NUL NUL NUL IEND
	http://image.baidu.com/search/down?tn=download&word=download&ie=utfffr=detail&u
	rl=http%3A%2F%2Fimg4q.duitang.com%2Fuploads%2Fitem%2F201501%2F13%2
	39_Ea5ry.jpeg&thumburl=http%3A%2F%2Fimg4.imgtn.bdimg.com%2Fit%2Fu%3D3348217047%
	2C21111/55801%26fm%2D21%26gn%2D0 ing
Normal text	t file length : 110,513 lines : 716 Ln : 1 Col : 1 Sel : 0   0 Windows (CR LF) ANSI INS

下载得到一张一模一样的图片



使用StegSolve左上角open打开下载出来的图(打开图片的顺序会影响结果,即1-2与2-1的区别)





操作<>小箭头对两张图片进行异或加减各种操作得到一张二维码



最后对二维码进行一个取反的操作,即将黑白两色取反,最后得到一个flag。

#### 4.LSB(最低有效位)

LSB替换隐写基本思想是用嵌入的秘密信息取代载体图像的最低比特位,原来的7个高位平面与替代秘密信息的最低位平面组合成含隐藏信息的新图形。

1.像素三原色(RGB)

2.通过修改像素中最低位的1bit来打到隐藏的效果

3.工具: StegSolve、zsteg、wbstego4、python脚本

# StegSolve

Analyse->Data Extract

File	Analyse Help
Norm	File Format
E	Data Extract
	Stereogram Solver
	Frame Browser
h	Image Combiner

打开之后选择RGB最低有效位

Bit Plane	es							
Alpha	7	6	5	4	3	2	1	0
Red	7	6	5	4	3	2	<b>1</b>	<mark>₽</mark> 0
Green	7	6	<u>5</u>	4	3	2	<b>1</b>	₽ 0
Blue	7	6	5	4	3	2	1	▶ 0
					https:	//blog.csdi	n.nei/weix	in_46555037

在右边Bit Plane Order选项栏中选择三基色的顺序,然后点击Preview得到flag



### zsteg

这个工具是在linux命令行上使用的

安装命令:

#### gem install zsteg

检测LSB隐写

zsteg xxx.png

root@kali:~/Deski	cop#_zsteg_model.png
imagedata	text: "#####\#\$\$\$##\#\$\$\$###"
b1, bgr, l <del>sb, xy</del>	toxt: "^Do u know how the bad man send secret messa
ges now? flag{bab	ylsbstego cannot be easier!%^%^&}\n" ←
b2, r, lsb, xy	file: Sview capture file
b2, r, msb, xy	file: VISX image file
b2,g,lsb,xy	file: 5View capture file
b2,g,msb,xy	file: VISX image file
b2,b,lsb,xy	file: 5View capture file
b2,b,msb,xy	file: VISX image file
b2,rgb,lsb,xy	file: 5View capture file
b2, rgb, msb, xy	file: VISX image file
b2,bgr,lsb,xy	file: 5View capture file
b2,bgr,msb,xy	file: VISX image file
b4,r,msb,xy	text: ["w" repeated 9 times]
b4,g,msb,xy	text: ["w" repeated 10 times]
b4,b,msb,xy	text: ["w" repeated 9 times]
b4, rgb, msb, xy	text: ["w" repeated the // hind bedn net/weivin /6555037
b4.bgr.msb.xv	https://biog.csun.net/weixin_4000007

#### 5.TweakPNG

TweakPNG是一款简单易用的PNG图像检查工具,它允许查看和修改一些PNG图像文件的元信息存储。

#### 使用场景: 文件头正常却无法打开文件,利用TweakPNG查看CRC值有没有错

打开TweakPNG

	TweakPNG	Ontions Tools Help			
	Chunk	Length CRC	Attributes	Contents	
	-				
File–>o	pen打开你要检查	的图片			
如果错i	误则会弹窗				
War	rning				×
		orrect crc for IH	IDR chunk (is cbd6	idf8a, should be c758d77d)	
				确定 https://blog.csdn.nel/weixin_4655	] 15037

知道CRC错误了之后用十六进制编译器打开这张图片,Ctrl+f找到提示的错误CRC然后把错误的值改好就可以打开图片了。

第二种情况就是CRC值没有出现错误,但是图片的高度或者宽度发生了错误,也会打不开图片,或者图片显示不完全,这是就 需要通过CRC计算出正确的高度或者宽度。

例: BUGKU里的一道题目,首先下载到一张图片







通过notepad++把CRC值改正之后再打开发现并没有得到Flag,说明了是这张图片的宽高设置有问题。

用python脚本通过CRC值计算出正确的宽高。



再打开图片对比查看

🤒 010 Edite	or - D:\for	kali\tm	pp\24-	25\3.p	ng							
File Edit	Search	View	Form	at So	ripts	Temp	lates	Tools	Wind	low I	Help	
i 🗋 🗸 📂	~ 🗖	۵	E 🖆	11	b 🗅	<b>1</b>	9 (9	L 🔎	AB	🦻 🔶	I A	
Startup	2. png	sim.	jpg	hexd	ata. rar	- 1	l.gif	3. p	ng×			
∓ Edit As	: Hex 🗸	Run S	cript ~	Rur	Templ	ate 🗸						
	Q	1	2	3	4	5	6	7	8	9	A	]
0000h:	: 89	50	4E	47	0D	0A	1A	0A	00	00	00	0
0010h:	00	00	01	F4	00	00	01	A4	08	06	00	0
0020h:	: 8A	00	00	00	09	70	48	59	73	00	00	1
0030h:	: 74	01	DE	66	1F	78	00	00	0A	4D	69	4
0040h	# L	) V I		on	P	115	2.1	VC			43	2
0050h		ſ .	3	33			, ,			-	58	9
0060h	500	) 5	50(	3							81	0
0070h			🔲 jr			_					12	4
0080h	hex	( (	JX	1f	4_	0>	(1-	F4			0A	4
0090h	1		n	nnn							38	E
00A0h			1.	utaues	http	s://blo	og.csd	n.net/	weixi	n_465	5503	$7_{\overline{E}}$

发现高的那段有错误 00 00 01 A4 ---> 00 00 01 f4

修改正确之后保存文件, 打开图片发现底部有flag



#### **6.Bftools**

Bftools用于解密图片信息。

使用场景:在windows的cmd下,对加密过的图片文件进行解密

格式:

Bftools.exe decode braincopter 要解密的图片名称 -o 输出文件名

Bftools.exe run 上一步写的输出的文件名

```
D:\CTF\bftools\bftools>bftools.exe decode braincopter zzzzzyu.png --output 123
png
```

D:\CTF\bftools\bftools>bftools.exe run 123.png XDCTF{ji910-dad9jq0-iopuno} ¤∳§∬¤⇔ D:\CTF\bftools\bftools>

#### 7.SilentEye

SilentEye是一款可以将文字或者文件隐藏到图片的解密工具。

使用场景:windows下打开SilentEye工具,对加密的图片进行解密

例:

打开SilentEye—>open 打开你想解密的图片



打开图片之后选择左下角decode解密



#### 点击Decode

S Decode message	e: D:/forkali/tmpp/27/SilentEye.bmp		?	×
	-Options	Image's encoding format :	HIP	- 🦷
	Image quality: 96.875% normal	<b>.</b>	Advar	loed
CharSet: VIF8 💌	Encrypted data F Compressed	av//blog.csdn.ne/ @Cancel 1	ے۔ De	

把生成的flag.txt文件保存在你想保存的地方

: D:/forkali/tmpp	/2//SilentEye.omp				?	2
- Options		Isage's	encoding	fermat :	BMP	•
Image quality:	96.875% normal	•			Adva	nce
Decoded file						0
	flag tet					
	flag. txt					
	-Options Inage quality: Decoded file	Options Image quality: 96.875% normal	Options Image quality: 96.875% normal	Options Image 's encoding Inage quality: 96.875% normal	Image's encoding format : Options Inage quality: 96.875% normal	Image's encoding format : BMP Options Inage quality: 96.875% normal Adva Decoded file Ileg txt

			省称		
			a flag.	txt	
			Siler	ntEye.bmp	
🗐 flag.t	bxt - 记事	本			
文件(F)	编辑(E)	格式(0)	查看(V)	帮助(H)	
flag{	This	isF_	lag}		
			-		

如果解密的时候需要密码,勾选encrypted data,输入密码和确认密码,再点击decode解密

	Opticas		Image's encoding	fernat :	ns • á
	Image quality: 9	. 8758 [soreal			hivanced
Grand States and State	Ley	ata 🗹 Compressed data		Cancel	) Decode

#### 8.Stegdetect

Stegdetect程序专门用于分析JPEG文件,如果确定了这个文件肯定使用了jpg的加密方式就可以用Stegdetect。因此用 Stegdetect可以检测到通过JSteg、JPHide、OutGuess、Invisible Secrets、F5、appendX和Camouflage等这些隐写工具隐藏的 信息。

格式:

stegdetect xxx.jpg

stegdetect -s 敏感度 xxx.jpgexi

thinking@ubuntu:~/Desktop\$	stegdetect	123456.jpg
123456.jpg : f5(***)		
thinking@ubuntu:~/Desktop\$	stegdetect	angrybird.jpg
angrybird.jpg : outguess(o)	ld)(*)	
thinking@ubuntu:~/Desktop\$	stegdetect	Pcat.jpg
Pcat.jpg : negative		
thinking@ubuntu:~/Desktop\$	stegdetect	-s 10.0 Pcat.jpg
Pcat.jpg : jphide(*)		
thinking@ubuntu:~/Desktop\$		

#### 压缩文件处理

#### 1.伪加密

如果压缩文件是加密的,或文件头正常但解压错误,首先尝试文件是否伪加密。zip文件是否加密是通过标识符来显示的,在每 个文件的文件目录字段有一位专门标识了文件是否加密,将其设置成00表示该文件未加密,如果成功解压则表示文件为伪加 密,如果解压出错说明文件为真加密。

#### 使用场景: 伪加密文件

操作方法:使用十六进制编译器打开压缩文件,找到从文件头开始数的第九第十个字符,将其修改为0000.

(1) Zip

1.十六进制编译器打开文件Ctrl+f搜索16进制的504B0102,可以看到每个加密文件的文件头字段。

75	79	61	6F	67	65	69	78	75	er.	ъE	72	69	SP	75	0.0
61	6E	64	61	6F	21	21	21	50	4B	01	02	1E	03	14	00
01	00	38	00	OB	8D	73	49	D8	15	C1	EF	46	67	00	00
0.0	AA	20	00	04	00	18	00	00	00	00	00	00	00	00	00
F6	81	00	00	00	00	31	2E	70	79	55	54	05	00	03	05
1	20	=0	-	20	0.0	00	0.4	0.4	00	00	00	00	<b>a i</b>	00	00

第九第十位为00 00的话是无加密,如果是00 01或者01 00的话是伪加密,将00 01改成00 00后解除伪加密状态。

(2) RAR

RAR文件由于有头部检验,使用伪加密时打开文件会出现报错,使用十六进制编译器修改标志位后如报错消失且正常解压缩, 说明是伪加密。使用十六进制编译器打开RAR文件,找到第24个字节,该字节位数为4表示加密,0表示无加密,将尾数改为0即 可破解伪加密。

#### 2.暴力破解

通常我们可以使用ARCHRP.exe工具来破解zip文件

#### windows下加密过的zip文件

1.攻击类型选择暴力破解,在范围位置根据提示选择暴力破解范围选项设置暴力破解包含的类型、开始于和结束于选项具体范围,如果没有定义则全范围暴力破解。点击打开选择要破解的文件,点击开始进行破解。建议使用1~9位的数字密码,以及系统自带的英文字典作为密码字典。

2.攻击类型选择掩码可以进行复杂的暴力破解,比如知道密码前3位是abc,后三位是数字,则在攻击类型选择掩码,在掩码出输入abc???,暴力范围选择所有数字,打开要破解的文件,点击破解。此时???的部分会被我们选择的暴力破解范围中的字符 代替。

ARCI	HPR 4.54 P	rofessiona	l Edition				
文件(F)	恢复(R)	帮助(H)					
<b></b> 約 約 一	· <i>解</i> 开始!	<b>停止</b>	③ 基准测试	<b>参</b> 升级	<b>?</b> 帮助	① 关于	退出
加密的 D:\CTF	ZIP/RAR/AC	E/ARJ 文件		攻击势 摘码	料理		•
范围 二月 二月 二月 二月 二月 二月 二月 二月 二月 二月 二月 二月 二月	长度 字	典 明文 Z(A - Z) Z(a - Z) I@) F	自动保存 ∃	送项 开始于: 詰束于: 撞码:	高级 abc???	用户定义	
状态窗 2016/1 2016/1 2016/1 2016/1	2/5 22:25:13 2/5 22:25:13 2/5 22:25:13 2/5 22:25:13 2/5 22:25:13	3 - 文件"D: \( 3 - 开始掩码 3 - 口令已成 3 - 'abc111' <del>\</del>	TF\rar\rar.zi 政击 政康复! 是这个文件的	p"已打开 的一个有	F。 效口令		* •
当前口令 已用时间 进度指示	): 【器		0.0	平均速 剩余时	度: 间:		
ARCHPR V	ersion 4.54	(c) 1997-201	2 ElcomSoft (	Co. Ltd.	niip <del>st//</del> btogres	dmnet/weixin	_48555037

破解成功后会提示(口令就是密码)

Advanced Archi	ve Password Recovery 统计信息:	
总计口令	111	
总计时间	15ms	
平均速度(口令/秒)	7,400	
这个文件的口令	abc111	6
十六进制口令	61 62 63 31 31 31	6

#### 3.明文攻击

明文攻击指的是,加密的ZIP文件中你已经知道了这个ZIP加密文件中的其中一个或多个文件,李永忠和谐内容推测出密钥并解密 ZIP文件的攻击方法,相比于暴力破解,这种方法在破解密码较为复杂的压缩包时效率更高。例如:假设一个加密的压缩包中有 两个文件readme.txt和flag.txt,其中flag.txt的内容是我们希望知道的内容,而我们拥有readme.txt这个明文文件,这时候就可以使 用明文攻击了。

操作:

1、将readme.txt的明文文件进行压缩,变成readme1.zip[#明文文件的压缩算法一定要和加密的压缩文件的算法一样!!不然会出错!

2、打开ARCHPR,攻击类型选择明文,明文文件路径选择readme1.zip即将明文文件不加密压缩后的文件)

3、选择要破解的文件,点击开始,破解成功后会获得密码。

图一

Anch	IPR 4.54 Pro	ofessiona	Edition				
5件(F)	恢复(R) ŧ	帮助(H)					
2	. 🔗	A	()	-	2		
打开	开始!	停止	基准测试	升级	帮助	关于	退出
加密的	ZIP/RAR/ACE	/ARJ 文件		攻击类型	!		
D:\CTF	baopo_qwb\b	aopo_qwb	o.zip	明文			
范围 明文选 明文文	长度   字典 项 件路径:	<b>則</b>	自动保存	选项	高級		
D:\CTF	\baopo_qwb\	readme 1.z	ip	- 💼	开始于		
2	密钥	-	密钥	-	密钥		- 🤮
	密钥 :使用二进制;	- 文件作为E	密钥 明文 ZIP 档案	- 文件	密钥		- 👙
■ 允许	密钥 :使用二进制; 口	- 文件作为F	密钥 明文 ZIP 档案	- 这件	密钥		•
六百年 一 允许 次态窗[ 2016/12 2016/12 2016/12 2016/12	密钥 :使用二进制; 2/5 23:30:20 - 2/5 23:31:12 - 2/5 23:32:24 - 2/5 23:32:36 -	- 文件作为F · 文件"D:\/ · 文件"D:\/ · 文件"D:\/	密钥 明文 ZIP 档案 CTF\baopo_q CTF\baopo_q CTF\baopo_q CTF\baopo_q	- 文件 wb\readme_ wb\readme. wb\readme. wb\readme.	密钥 jiami.zip"已 wb.zip"已打开 zip"已打开	打开。 1开。 。	- 49
<ul> <li>六许</li> <li>大许</li> <li>大次</li> <li>2016/12</li> <li>2016/12</li> <li>2016/12</li> <li>前日</li> <li>10</li> <li>10</li> <li>11</li> <li>11</li> <li>12</li> <li>12</li> <li>12</li> <li>14</li> <li>15</li> <li>15</li> <li>16</li> <li>16</li> <li>17</li> <li>16</li> <li>16</li> <li>17</li> <li>16</li> <li>17</li> <li>16</li> <li>17</li> <li>16</li> <li>17</li> <li>16</li> <li>16</li> <li>17</li> <li>16</li> <li>1</li></ul>	密钥 ·使用二进制 2/5 23:30:20 · 2/5 23:31:12 · 2/5 23:32:24 · 2/5 23:32:36 · :: :: :: 器	文件作为E 文件"D:\( 文件"D:\( 文件"D:\( 文件"D:\(	密钥 明文 ZIP 档案 CTF\baopo_q CTF\baopo_q CTF\baopo_q CTF\baopo_q	· 文件 wb\readme_ wb\paopo_g wb\readme. wb\readme. 平均速度: 剩余时间:	密钥 jiami.zip"已 wb.zip"已打开 zip"已打开 zip"已打开	打开。 1开。 。	- 49

图二

ARCHPR 4.54 - 0%			2
文件(F) 恢复(R) 帮助(H)			
口令已成功恢复!		[	×
Advanced Archive Pas	ssword Recovery 统计	·信息:	
总计口令	n/a		
总计时间	3m 32s 608ms		
平均速度(口令/秒)	n/a		
这个文件的口令	111111		0
十六进制口令	31 31 31 31 31 31 31		3
<b>日</b> 保存…	•	确定	-
□ 允许使用二进制文件作为明文 ZIP	档案文件		
状态窗口			
2016/12/5 23:35:37 - 明文攻击已开始 2016/12/5 23:39:10 - 加密密钥已成功 2016/12/5 23:39:10 - 口令已成功恢复! 2016/12/5 23:39:10 - '111111' 是这个文	恢复! 【件的一个有效口令		* •
当前口令: n/a 已用时间: 3m 32s 明文攻击正在进行, 尝试找回口令(最长	平均速度: 剩余时间: ミ9个符号) 0%	n/a 49d 17h 2m 47s	
ARCHPR version 4.54 (c) 1997-2012 ElcomS	Soft Co. Ltd. https://blo	g.csdn.net/weixin_46	1555037

有时候不一定能破解出文件口令,但是能够找到加密密钥等信息,可以直接将文件解密,点击确定保存解密后的文件即可。 还有一种情况就是加密的文件无论如何都是解密不了的,通过破解软件可以得到它的加密密钥,Flag就在加密密钥上



#### 流量取证技术

流量取证题目通常都是给出一个pcac文件,需要用wireshark对这个pcac文件进行一个流量检测。

过滤ip地址

如源IP或者目标X.X.X.X

ip.src eq x.x.x.x or ip.dst eq x.x.x.x 或者 ip.addr eq x.x.x.x

过滤端口

tcp.port eq 80 or udp.port eq 80

tcp.dstport == 80 只显示tcp协议的目标端口为80

tcp.srcport == 80 只显示tcp协议的源端口为80

tcp.port >=1 and tcp.port <= 80

过滤协议

直接tcp/udp/arp/icmp/http/ftp/dns/ip...

#### 过滤MAC

eth.dst == A0:00:00:04:C5:84

包长度过滤

udp.length == 26 # 这个长度是指udp本身固定长度8加上udp下面那块数据包之和 tcp.len >= 7 # 指的是ip数据包(tcp下面那块数据),不包括tcp本身 ip.len == 94 # 除了以太网头固定长度14,其他都算是ip.len,即从ip本身到最后 frame.len == 119 # 整个数据包长度,从eth开始到最后 http模式过滤 http.request.method == "GET" # 筛选http协议里面的GET请求方法 http.request.method == "POST" # 筛选http协议里面的POST请求方法 http.request.uri == "/img.logo-edu.gif" # http contains "GET" # 筛选http协议里面包含GET字段的包 http contains "HTTP/1." # 筛选http协议里面包含HTTP/1字段的包 http.request.method == "GET" && http contains "User-Agent." http contains "flag" # 筛选http协议里面包含flag字段的包 http contains "key" # 筛选http协议里面包含key字段的包 TCP类的也可以这样:

tcp contains "flag" # 筛选TCP协议中有包含flag字段的包

#### WireShark协议分析

解析:在出题人的角度看,出题人在制作这个流量包的时候肯定是先写好过滤代码—>点击开始抓包—>发送图片—>停止抓包。 所以做出来的流量包绝大部分应该是与flag相关的,只有少部分是与flag无关的没来得及剔除的无关流量,所以使用一个协议分 级能很好的找出与flag相关的包。

1.点击统计—>协议分析

统计(5)	电话凹	无线(W)	工具①	帮助(出)	
捕る	这件属性		Ctrl+Alt+Shift+C		
已第	解析的地址				
协议	2分级(巴)				

2.分析哪些包占比大

协议	按分组百分比	分组	按字节百分比	字i
✓ Frame	100.0	1083	100.0	884
✓ Ethernet	100.0	1083	1.7	151
> Internet Protocol Version 6	0.6	7	0.1	906
✓ Internet Notocol Version 4	99.4	1076	2.4	215
> User Datagram Protocol	3.0	32	0.0	256
> Transmission Control Protocol	96.4	1044	https://blog.csdf.net/weixin_465	844

比如这个 IPv6很少 IPv4很多,在IPv4里面又有UDP和TCP,TCP占大多数。

#### WireShark流汇聚

再按关注的http数据包或tcp数据包中选择流汇聚,可以将HTTP流或TCP流汇聚或还原成数据,在弹出的框中可以看到数据内容。

	-1								
No.	Time	Source	Destination	Protocol Leng	th	Leftove: Info	(	- UTTD /4 4	(lienties (
	5 0.0000/5	192.108.1/9	192.108.1/9.2		07	PUST	/uploads/security.pr	np HIIP/1.1	(application/x-www-form-urlencoded)
-	10.002025	192.100.1/9	192.100.1/9.2		120	DOG	标记/取消标记 分组(M)	Ctrl+M	(and) (and (a cost from unlanded)
Ť	10 4.259582	192.108.1/9	192.108.179.2		07	PU:	忽略/取消忽略 分组(1)	Ctrl+D	(application/x-www-form-urlencoded)
	12 4.202/00	192.100.1/9	192.100.1/9.2		197	DOG	设置/取消设置 时间参考	Ctrl+T	(analization /
	13 0.43/202	192.100.1/9	192.100.1/9.2		07	PU	时间平移	Ctrl+Shift+T	(application/x-www-form-urlencoded)
	1/ 0.441240	192.108.179	192.100.1/9.2		197	POG	分组注释	Ctrl+Alt+C	(and) (and (a see from a located)
	25 11.295/78	192.168.179	192.108.1/9.2		007	PU			(application/x-www-form-urlencoded)
	26 11.312370	192.168.179	192.168.179.2	HTTP 3	97	HII	编辑解析的名称		
	29 11. 572/10	192.108.1/9	192.108.1/9.2		01	PU	作为过滤器应用	,	(application/x-www-form-urlencoded)
	31 11.3/5/94	192.108.1/9	192.108.1/9.2	. HITP 5	91	HII	准备讨法器		
							「日本」の「二日本」の中国家		
							NIG CONTRACT		
							刘诺看巴	,	
							SCTP	,	L
> 1	rame 5: 834 byt	es on wire (6	572 bits), 834 H	bytes capture	ed (	6672 bits	追踪流	•	TCP 流
> 6	thernet II, Sro	: Vmware_7c:7	3:6c (00:0c:29:	7c:73:6c), Ds	st:	Vmware_de	使制		UDP 流
> 1	Internet Protoco	l Version 4,	Src: 192.168.17	9.246 (192.16	58.1	79.246),	06.90		SSL 流
~ 1	ransmission Con	trol Protocol	, Src Port: 500	07 (50007), [	)st	Port: htt	协议首选项	•	WHTP And antiweivin 48555087
	Source Port:	50007 (50007)					解码为(A)		
	<pre>st: set2.mail.qq. onnection: keep-al ontent-Length: 143 oche-Control: no-c rigin: http://set2</pre>	com ive ache, max-age=0 .mail.qq.com							^
I	If-Modified-Since: 0								
C	User-Agent: Mozilia/5.0 (Windows NI 6.1; WUW64) AppleWebkit/537.36 (KHIML, like Gecko) Chrome/33.0.1/50.146 BiDUBrowser/6.x Safari/537.36 Content-Type: application/x-www-form-urlencoded								
A	Accept: */*								
Re	Referer: http://set2.mail.qq.com/zh_CN/htmledition/ajax_proxy.html?mail.qq.com&v=140521								
A	Accept-Encoding: gzip,detlate								
Co	Cookie: ssuid=9979081647; ptui_loginuin=81101652; o_cookie=81101652; pgv_pvid=3703132940; newpt=2;								
pt	cz=4d9c0097882e7b	9300db96ff8272c	602d390465e21f8cc	52d4a4719b7a73	dfff	f; pt2gguin=0	0081101652; uin=0008110	01652; skey=@z	TisQEDyk; p_uin=00081101652;
P_	<pre>p_skey=SMR2Xte-5d352j-LIjK01POC1XsLYf6J5WR70YT1x1s_; pt4_token=VqHcOafacF-cGmaHbpHVyg_; wimrefreshrun=08; qm_flag=0; qqmail_alias=81101652@qq.com; </pre>								
q	<pre>sid=sild(b)2&amp;l8C4949ed/syb41d6d)2e/3949a34d959a_q0012M1h0/25112/UNIWmotFL1q508XUE9Ubth21FLmNko1V113KFLU0btgxc18.; qm_username=81101652; om_sid=18e/a549a94041d8A5273049a54d969a_d1015M1h0/25112/DNIWmotFE1c58xUE9Ubth21FLmNko1V113KFLU0btgxc18.; qm_username=81101652;</pre>								
q	<pre>qm_tsk=81101652&amp;@zTisQEDyk; foxacc=81101652&amp;0; ssl_edition=mail.qq.com; edition=mail.qq.com; username=81101652&amp;81101652; (CSHOW=000001;</pre>								
ne	w_mail_num=811016	52&0; webp=1; p	tisp=cn						
{"path":"fly.rar", "appid":"", "size": 525701, "md5": "e023afa4f6579db5becda8fe7861c2d3", "sha": "ecccba7aea1d482684374b22e2e7abad2ba86749", "sha3":""}HTTP/1.1									
Content-Type: application/json; charset=GB18030									
Co	ontent-Length: 100	3							
E.	{"result":{"errCode":0,"message":""},"data":{"sIP":"sz.mail.ftn.qq.com","nPort":								
8	80, "sKey": "7cb0f705d9067792b5023cb028ba442e653338521c5067213019b0958608c4060b3e756c14212ef37e7f876f7c815af7fc0463df85f921a63efbe4b4f14ee12193de81cb69be1b8								
b4	p4p591a37b37b37b37b37b37b37b37b37b37b37b37b37b3								
91	at2a3baa7b9t68850	0747fd5d1079c0c	a5c9d371fe350b8f4	3c7b7a8e07fa1f	2000	326869768ef	1e77021e2d8811f0fc2dd71	133d48f3301ba1	edide/aaaa5//DCD9d1ctDta1ett/639aCe44 57403b8bbab91264c5c1e3e46007cb4d3e8a5
7	735", "sFileId": "ypsPlpS%2B0dcfe5jb0tzVcj3jXCmQKVim%2FapXkA%2ByRCaR%2Bkzg%2Ftyqzfnbez%2FqnwRwH								

%2Fx0AhBWfJCRFBof7pxgXx97Agu1gA 1424837275,"exist":0,"sMailId":	<pre>EezCOgDVHfVT5XMqwFGhipxAhZU1hRrFdv", "ZF0025-Gk7VNrX_mE2gJSF1vKH1Y52"}}</pre>	nAppId":"", "nDownCnt":0, "nCreateTime":1424837275, "nExpireTime":0, "nServerSecond":
1 客户端 分起, 1 服务器 分起, 1 turn(s).		() <u>*</u>
Entire conversation (2689 bytes)	•	亞不和保仔類語为 ASCII ·
查找:		查找下一个())
		渡掉此流 打印 Save agw 0.00. Back 1.10 W Close _ 400 K Lp 37

红色的是文件头,蓝色的正文内容,通常flag都藏在正文内容中。

#### 常见的HTTP流关键内容:

1.HTML中直接包含重要信息。

2.上传或下载文件内容,通常包含文件名、hash值等关键信息,常用POST请求上传。

3.一句话木马,POST请求,内容包含eval,内容使用base64加密。