MISC入门总结



<u>想成菜鸡的武阳</u> 已于 2022-04-27 17:00:23 修改 2章标签: <u>安全</u> 于 2022-04-19 20:00:01 首次发布 版权声明:本文为博主原创文章,遵循 <u>CC 4.0 BY-SA</u>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/weixin_53268624/article/details/124276230</u> 版权 《re入门到Misc精通》,哄堂大笑了,家人们。

总结在后面

ctf.show Phttps://ctf.show/challenges#misc2-1134

ctfshow misc入门题(还没写完,持续更新)

图片篇第一题 签到题 打开直接给flag qq提取文字交了

第二题 给个txt文件 winhex打开txt文件后 发现

	misc2.txt																					
٦	Offset	0	1	2	3	- 4	5	6	- 7	8	- 9	A	В	C	D	Е	F					,
	00000000	89	50	4E	47	OD	ΟA	1A	ΟA	00	00	00	OD	49	48	44	52	∎PNG		I	HDR	
	00000010	00	00	03	84	00	00	00	96	08	06	00	00	00	86	B8	46	I	н. Т		I,F	
	00000020	36	00	00	00	01	73	52	47	42	00	AE	CE	1C	Ε9	00	00	6 s	RGB	®Î	é	
	00000030	00	04	67	41	4D	41	00	00	Β1	8F	ΟB	FC	61	05	00	00	gAMA	ι ±	üa		
	00000040	00	09	70	48	59	73	00	00	12	74	00	00	12	74	01	DE	pHYs	; t		tÞ	
	00000050	66	1F	78	00	00	1B	F5	49	44	41	54	78	5E	ED	DD	ЗB	fx	õIDA	Tx^	íÝ;	
	00000060	72	DC	38	Β7	CO	71	F8	AE	45	72	30	E5	15	Β4	57	281	R ^Ü 酱粮	199EF	DÅ:	EKKEI	
	00000070	3B	99	68	D2	C9	Α4	DO	4E	9C	7D	Α1	33	27	52	28	65	;∣hÔĔ ²	ÐNI }	131	R(e	

png头 所以 换成png后缀 打开后给flag 文字识别 提交

第三题 bpg格式 cmd进入查看bgp文件中 命令为: bpgview.exe E:\ctfshow\misc3.bpg

第四题 给了四个txt winhex进入发现后缀不对 改了后每个图 对应一部分 都改成png 就可以看到了

第五题 打开图片 发现假flag winhex找到后面连接的flag

第六题 打开winhex 搜文字 ctfshow即可得到

第七题:同上

第八题: flag在图片文件中图片文件中。 kali用foremost命令分离出来 藏了一个png 开始标志 89 50 结束标志 60 82

第九题:同第七题



第十题:

第十一题: flag在另一张图里

PNG中IDAT

图像数据块IDAT(image data chunk): 它存储实际的数据,在数据流中可包含多个连续顺序的图像数据块。 binwalk看到的zlib就是其压缩格式 binwalk -e可以实现自动解压

Startup	r	nisc	11.p	ng	m	ISCI	1.pn	ıg ×									
	0	1	2	3	4	5	6	7	8	9	Α	В	С	D	Е	F	0123456789ABCDEF
0000h:	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	‰PNGIHDR
0010h:	00	00	03	84	00	00	00	96	08	02	00	00	00	09	DA	D1	"–ÚÑ
0020h:	61	00	00	0B	73					78	9C	ED	DD	3D	7A	ΕA	asIDATxœíÝ=zê
0030h:	46	1B	06	60	F9	5B	8B	9D	E2	5C	59	01	5E	01	4E	93	F`ù[‹.â\Y.^.N"
0040h:	2A	6D	ЗA	28	ED	26	DD	29	D3	A5	31	A5	DD	A5	4D	95	*m:(í&Ý)Ó¥1¥Ý¥M•
0050h:	26	B 0	02	9F	15	E4	ЗA	45	CC	5E	F8	0A	30	1A	81	10	&°.Ÿ.ä:EÌ^ø.0
0060h:	(02)	0B	BD	92	B8	EF	EA	FC	60	34	8C	24	CF	C3	48	7A	½′,ïêü`4Œ\$ÏÃHz
0070h:	E7	66	B5	5A	65	00	00	10	E1	7F	D1	0D	00	00	E0	7A	çfµZeá.Ñàz
0080h:	09	A 3	00	00	84	11	46	01	00	80	23	8C	02	00	10	46	.£".F#ŒF
0090h:	18	05	00	20	8C	30	0A	00	40	18	61	14	00	80	30	C2	ŒO@.a€OÂ
00A0h:	28	00	00	61	84	51	00	00	C2	80	A3	00	00	84	11	46	(a"QÂ.£".F
00B0h:	01	00	80	23	8C	02	00	10	46	18	05	00	20	8C	30	0A	#ŒF ŒO.
00C0h:	00	40	18	61	14	00	80	30	C2	28	00	00	61	84	51	00	.@.a€OÂ(a"Q.
00D0h:	00	C2	80	A3	00	00	84	11	46	01	00	80	23	8C	02	00	.Â.£".F#Œ
00E0h:	10	46	18	05	00	20	8C	30	0A	00	40	18	61	14	00	80	.F Œ0@.a€
00F0h:	30	C2	28	00	00	61	84	51	00	00	C2	80	A 3	00	00	84	0Â(a"QÂ.£"
0100h:	11	46	01	00	08	23	8C	02	00	10	46	18	05	00	20	8C	.F#ŒF Œ
01106.	20	۸	00	10	10	<u>c</u> 1	1 /	00	00	эn	CO	20	00	00	<u>c</u> 1	0 /	
Templa	te R	esu	lts -	PN	G.b	t æ											

Name	Value	Start	Size	Color	Comment	
> struct PNG_SIGNATURE sig		0h	8h	Fg: Bg:		
> struct PNG_CHUNK chunk[0]	IHDR (Critical,	. 8h	19h	Fg: Bg:		
v struct PNG_CHUNK chunk[1]	IDAT (Critical,	. 21h	B7Fh	Fg: Bg:		
uint32 length	2931	21h	4h	Fg: Bg:		
> union CTYPE type	IDAT	25h	4h	Fg: Bg:		
> ubyte data[2931]		29h	B73h	Fg: Bg:		
uint32 crc	C464AE32h	B9Ch	4h	Fg: Bg:	CSDN @想	成茲鸡的武阳
> struct PNG_CHUNK_chunk[2]	IDAT (Critical	RA0h	1D81h	Fa: Ba:		

用010把第一个idat删除了 然后另存为图 即可求出

第十二题:同上发现idat过多删除前8个idat就好

第十三题:提示在后面发现

00000EE0	FC	DC	FΕ	33	D2	72	35	CO	72	BB	97	92	ΒE	5C	89	23	ü <mark>U</mark> þ30r5Ar» ′ ¾∖ #
00000EF0	88	Β8	53	8D	17	FЗ	F9	63	1A	74	Β9	66	85	73	86	68	∣,S óùc t¹f∣s∣h
00000F00	AA	6F	4B	77	BO	7B	21	61	14	65	53	36	Α5	65	54	34	ªoKw°{!a eS6¥eT4
00000F10	34	36	78	63	25	34	DD	38	EF	66	AB	37	10	33	95	39	46xc%4Ý8ïf≪7 3∎9
00000F20	1F	62	82	37	BA	65	45	62	7C	32	54	64	7E	31	ЗA	64	b 7ºeEb 2Td~1:d
00000F30	E 4	65	F 1	36	FA	65	F5	34	1 <mark>E</mark>	31	07	32	1D	66	54	CBBE	随鱼鱼肉和肉面加
00000740	17.1	22	22	20	ΕO	C 1	60	70	20	ΠE	πo	ЪE	25	A A	∇C	an	# 2204 - 1 1 · # 2 Ã \ D Í

查看发现 中间隔两个字符

s="631A74B96685738668AA6F4B77B07B216114655336A5655433346578612534DD38EF66AB35103195381 flag=""

for i in range(0,len(s),4):#相当于四个数字个循环,只要前两个

....

flag += s[i]

flag += s[i+1]

print(flag)

•

得到十六进制转换 16进制转换, 16进制转换文本字符串, 在线16进制转换 | 在线工具 (sojson.com)

第十四题: flag在图片文件中图片文件中。 kali用foremost命令分离出来 藏了一个png 开始标志 89 50 结束 标志 60 82

Þ.

第十五题: winhex打开就发现了flag

第十六题: winhex打开时发现有大量IDAT块 提示flag在图片数据里

有zlib文件 见第十题 binwalk -e分离 得到flag

第十七题: zsteg,这是一个用于检测PNG和BMP中的隐藏数据隐藏数据的工具,可以快速提取隐藏信息 (86条

消息) Kali linux下图片隐写,图片隐写信息快速检测工具——zsteg_老魏一凡的博客-CSDN博客

ops_request_misc=&request_id=&biz_id=102&utm_term=kali%E5%A6%82%E4%BD%95%E8%A3%85zsteg% task-blog-2~all~sobaiduweb~default-0-

116953571.142%5Ev9%5Econtrol,157%5Ev4%5Econtrol&spm=1018.2226.3001.4187

4 genis In	stat	rea																
root@kali	:~# 2	zste	eg -	-a /	roo	ot∕¦	桌面	/m:	isc1	7.pr	۱g							
00000	000:	e1	1f	30	53	86	4f	c5	a4	1b	f5	e6	e5	с7	46	Øa	92	ØS.O
00000	010:	9b	ee	72	e7	с9	9e	b9	a7	74	de	92	4d	ad	61	5b	58	r
00000	020:	f2	98	65	77	2b	d2	d3	85	32	fc	08	83	86	1f	Øf	1e	ew+
00000	030:	cb	ab	ac	9c	4b	са	02	20	e2	ce	e4	ae	60	1a	2c	c6	к
00000	040:	7b	c8	9a	77	31	2f	9e	67	db	d9	3e	53	fe	17	a5	50	{w1/.g
00000	050:	20	e5	1d	8c	d5	49	4e	52	a5	54	31	cb	8b	c5	3b	09	INR
00000	060:	a2	a6	fe	5b	da	4f	9e	78	9c	5d	46	d6	e2	6b	6b	2a	[.0.x
00000	070:	f2	62	0c	ba	70	19	a0	27	f3	84	77	99	02	77	05	79	.bp'
00000	080:	5b	44	b7	79	b3	54	11	a1	f3	54	34	56	7e	ff	55	d1	[D.y.T
00000	090:	c6	39	90	c 8	21	7f	26	39	44	58	78	с3	ed	37	4a	7c	.9!.89
00000	0a0:	50	24	e8	79	7b	4b	9c	fa	2a	2c	bb	e8	b9	fb	40	2ç	P\$.y{K
00000	0b0:	50	05	21	4c	3b	29	65	b4	60	1c	27	bb	48	SPAN	節	小水采)	判的或的e.

....

zsteg -E /root/桌面/misc17.png 'extradata:0' > 1.txt

binwalk -e /root/桌面/1.txt 得到

第十八题:提示flag在标题、作者、照相机和镜头型号里 鼠标右键 属性

常规	安全	属性修改 详细信息 以前的版本
属性]	值 ^
标题		ctfshow{32
分级 标记		$\bigstar \div \bigstar \bigstar \bigstar$
备注 来派	₹	
作者 拍摄 程序	日期 名称	5d60c208f7 CSDN @想成菜鸡的武阳

National States and American State American States and American Stat American States and American Stat American States and American States and Am

•

第十九题: flag在主机上的文档名里 属性里没 EXIF信息查看器 (tuchong.com)

第二十题: flag在评论里 同上

Commont	这图片也太难看了。来自:西替爱抚秀大括号西九七九六四必一诶易西
Comment	爱抚零六易一弟七九西二一弟弟诶弟五九三易四日本插程成菜鸡的武阳

>

谐音梗扣钱 ctfshow{c97964b1aecf06e1d79c21ddad593e42}

第二十一题: flag在序列号上 同上 找到序列号一串数字

16进制转换文本/文本转16进制

686	657	828	582	659	7329

字符串转16进制 >>

hex(X&Ys)

CSDN @想成菜鸡的武阳

python中hex是把十进制转十六进制 把x的分辨率与Y的分辨率再相加

十进制转十六进制| 10进制转16进制 | 在线进制转换 (sojson.com)

X分辨率	3902939465
Y分辨率	2371618619
PageName	https://ctf.show/
X定位	1082452817
Y定位	2980145261
目标Printer	ctfshow{}
	CSDN @想成墓鸡的武阳

四个值转换后相加 即可得到flag

3902939465		转换		
------------	--	----	--	--

进制	结果。
二进制	1110100010100010001000
四进制	32202202011021
八进制	35050420511
十进制	3902939465
十六进制	e8a22149 CSDN @想成菜鸡的武阳

ctfshow{e8a221498d5c073b4084eb51b1a1686d}

第二十二题:

提示 flag在图片里 winhex打开后搜索ctfshow 没有发现

学习到了一个新的jpeg压缩 我好菜啊 MagicEXIF 元数据编辑器_官方电脑版 _华军纯净下载 (onlinedown.net)

INIAYICEAN ISINKAJAJAAA

	{there_is_no_flag_here}	
300 × 50, 100% [▲] 「「 缩略图 (IFD1) ▼ 【	CSI	DN @想成菜鸡的武阳

ctfshow{dbf7d3f84b0125e833dfd3c80820a129}

第二十三题: flag在时间里

给了一个psd文件

EXIF信息查看器无需安装软件,只需上传照片即可查看完整EXIF信息,包括机身、镜头型号、拍摄时间、相机 快门次数,支持JPEG、TIFF、CR2、NEF、XMP等多种图片格式。无需下载,比Exif Show, ExifPro更好用的

EXIF查看器!

https://exif.tuchong.com/查看时间

XMP-photoshop

色彩模式	RGB
TextLayerName	{there is no flag here}
TextLayerText	{there is no flag here}

XMP-xmp

创建日期	2021:03:25 15:45:24+08:00	
Creator工具	Adobe Photoshop CC 2019 (Windows)	
元数据Date	2021:03:25 16:02:50+08:00	
修改日期	2021:03:25 16:02:50+08:00	CSDN の相応装速的

	奴据 (F:) ◇ ctf工具集 ◇ misc ◇ 图像隐写 ◇ 垃圾	⟨堆大量工具 〉 exiftool-	18.25		~ Ū	。 た exiftool-18.2
从大一下学期期⇒^	名称 ^ ^	修改日期	类型	大小		
	🙀 exiftool.exe	2016/8/3 7:54	应用程序	7,063 KB		
ŧ.	🖹 misc23.psd	2021/3/25 16:33	Adobe Photosh	65 KB		
- Personal C:\U:	understand sers\武阳>cd/ f: cd F:\ctf工具集\misc\图]像隐写\垃圾	堆 大量工具	Veriftool-18.25		
F:\c Exif File Dire File File File	tf工具集\misc\图像隐写\ Tool Version Number Name ctory Size Modification Date/Time Access Date/Time Creation Date/Time	垃圾堆 大量 : 10.2 : misc: : . : 64 kl : 2021 : 2022 : 2022	工具\exiftc 5 23.psd B :03:25 16:1 :04:26 15:1 :04:26 15:1 w-rw-	001-18.25>exiftool mis 33:07+08:00 57:54+08:00 57:54+08:00	sc23. psd	

nistory netron	. CUSHOW (), UNIXITIES tamp, DECIDIEA, getting
History Instance ID	: xmp.iid:1, xmp.iid:2, xmp.iid:3, xmp.iid:4
History Software Agent	: Adobe Photoshop CC 2019 (Windows), Adobe Photoshop CC 2019 (Windows), Adobe Photoshop CC 201
9 (Windows), Adobe Photoshop (C 2019 (Windows)
History When	: 1997:09:22 02:17:02+08:00, 2055:07:15 12:14:48+08:00, 2038:05:05 16:50:45+08:00, 1984:08:03
18:41:46+08:00	CSDN @想成菜鸡的對應

可以看到action 和history

要求是先转成unix时间戳 然后DEC十进制转HEX十六进制就得到flag

时间戳(Unix timestamp)转换工具 - 在线工具 (tool.lu) Phttps://tool.lu/timestamp/

时间	1997-09-22 02:17	北京时间	转换»	874865822	秒(s) ~
					CSDN @想成菜鸡的武阳

十进制转十六进制| 10进制转16进制 | 在线进制转换 (sojson.com) https://www.sojson.com/hexconvert/10to16.html

○ 2进制 ○ 8进制 ◉ 10进制 ○ 16进制 ○ 32进制 ○ 64进制 | 更多进制:

10 ~

步骤:上面选择当前进制,然后下面输入数值,再点【转换】按钮,就能得到常见的进制数

转换

进制	结果
二进制	110100001001010100100
四进制	310021112102132
八进制	6411262236
十进制	874865822
十六进制	3425649e CSDN @想成菜鸡的武阳

合并就好 ctfshow{3425649ea0e31938808c0de51b70ce6a}

misc41:

太坑了 竟然是

🇱 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) 有	帮助(H)
---	-------

居		R 448 AR 7 TA 🔽 🔽 🗢 🍣 🧼 🔜 🎾 🕸 🖪 🚩	W
编辑(<u>D</u>)	misc41.jpg		
	位置管理器 (全部)		
	Offset▲ 搜索结果	时间	
	2A3A F001	2022/04/26 1	
	2A40 F001	2022/04/26 1	
	2A42 F001	2022/04/26 1	
	2A44 F001	2022/04/26 1	
	2A48 F001	2022/04/26 1	
	2A4A F001	2022/04/26 1	
	2A4C F001	2022/04/26 1	
	2A50 F001	2022/04/26 1	
	2A5A F001	2022/04/26 1	
	2A60 F001	2022/04/26 1	
	2A6A F001	2022/04/26 1	
	2A70 F001	2022/04/26 1	
	2A72 F001	2022/04/26 1	
	2A74 F001	2022/04/26 1	
	2A7A F001	2022/04/26 1	
	2A7C F001	2022/04/26 1	
	2A92 F001	2022/04/26 1	
	2A94 F001	2022/04/26 1	
	2A9A F001	2022/04/26 1	
	Offset 0 1 2 3 4 5 6 7 8 9 A	B C D E F 🔿 =	
	000029F0 02 8A 28 A0 02 8A 28 A0 02 8A 28	A0 02 8A 28 A0 I(I(I(I(
	00002A00 02 8A 28 A0 02 8A 28 A0 02 8A 28	A0 02 8A 28 A0 I(I(I(I(
	00002A10 02 8A 28 A0 02 8A 28 A0 02 8A 28	A0 02 8A 28 A0 I(I(I(I(
	00002A20 02 8A 28 A0 02 8A 28 A0 0F 4D C9	4E 9D 58 55 D8 I(I(MÉN XUØ	
	00002A30 B5 FD 47 69 53 D7 FF 5B 01 6A F0	01 01 E0 EE DF µýGiS×ÿ[jð àîß	
	00002A40 F0 01 F0 01 F0 01 EA 39 F0 01 F0	01 F0 01 87 55 8 8 8 698 8 8 1U	
	00002A50 F0 01 A3 B2 47 4B 4C F6 FC AC F0	01 EF C7 2D A1 8 £2GKLöu-8 ïÇ-1	
	00002A60 F0 01 84 80 67 39 B8 BF 67 8B F0	01 1E 8F AB 89 8 1g9,2g18 «	
	UUUU2A7U FU U1 FU U1 FU U1 EA UE A3 U3 FU	UI FU UI 6C 6U a a a e E a a I'	
	UUUU2A8U US SU UE 4D 31 A1 21 93 A2 F3 FB	UB D5 ED 4F UA P MITTE cou U1U	
	00002A90 D3 78 F0 01 F0 01 39 6D A4 58 F0		
	00002AA0 F5 AD F0 01 40 67 0D A4 F0 01 9E		
	00002AD0 F0 01 F0 01 F1 06 68 94 F0 01 F0		
	00002AE0 4B 41 41 C9 9B 0E E8 6A EB 73 E1	D2 76 58 11 48 KAAÉL èjësáÒvX J	
	00002AF0 F0 01 12 94 0A 13 24 01 FE 15 39	D1 56 68 9F 9A 8 S b 9NVh	
	00002B00 F0 01 2E 6B 3A 6F C1 F8 F0 01 F0	01 F0 01 D7 16 ð .k:oÁøð ð ð ×	
	00002B10 F0 01 F0 01 F0 01 CA D2 F0 01 4A	E6 F0 01 5E 9B 5 5 5 ÊÒ5 Jæ5 ^	
	00002B20 F0 01 EC 72 F0 01 DC 88 F0 01 16	27 <mark>FO O1 </mark> 3C 9A <mark>ð ìrð</mark> Ü ð ′ð <	
	00002B30 F0 01 66 62 F0 01 A2 EA F0 01 F0	<mark>01 FO 01</mark> 1E 6E <mark>ð</mark> fb <mark>ð</mark> ¢êð ð ð n	
	00002B40 F8 EE 08 C9 CA 06 EF 2D FE 04 73	2E B9 C2 AE E2 øî ÉÊ ï-þ s.¹Â®â	
	00002B50 F0 01 1A BA FE 30 CC 84 F0 01 82	1F <mark>FO 01 FO 01 ð</mark> ºþ0Ìl <mark>ð ð ð</mark>	CSDN @想成菜鸡的
	00000000 0 00 01 D0 54 D0 01 D5 00 D0 01 BD	OF FO OI OF 73 X ITX 18 N X V	



H4ppy Apr1I F001's D4y! F001

得到的是ctfshow{fcbd427caf4a52f1147ab44346cd1cdd}

misc24

flag在图片上面。

(9	0条消息) 位图	룊(bi	mp)	文化	+格	式分	→析	_ai	dem	n_bro	own	的博	專客	-CS	SDN	博	客_b	m	p位图
ht	tps://blog.csc	ln.n	et/a	ide	m_l	b ro \	vn/a	artio	cle/o	detai	ls/8	050	0063	37					
	Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
	00000000	42	4D	FO	4C	ΟA	00	00	00	00	00	36	00		局	28	00]	
7	7 00000016	00	00	84	03	00	00	96	00	00	00	01	00	18	00	00	00		
1	00000032	00	00	ΒA	4C	ΟA	00	12	ΟB	00	00	12	0B	00	00	00	00		
	00000048	00	00	00	00	00	00	FF	FF	FF	FF	FF	F_{C}	.FR	馬	野	胚		
	00000064	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	÷.	
										3	84	4							
	HEX 384																		
	DEC 000								CS	DN @楚	,成菜)	鸣	杜 卡		但至	Пф	40	იი	
1	DEC 900												1719	ビノロ	行き	小见	1731	00	

最后得出目前文件是900*150=135000个像素大小同时文件头占53字节

文件尾的位置在675053字节处(后面两个字节是windows的"补0"),又因为每个像素点由3个字节(十六进制码 6位)表示,每个字节负责控制一种颜色,分别为蓝(Blue)、绿(Green)、红(Red),所以文件真实的像素大小为:(675053-53)/3=225000

提示高 所以正确的高度是225000/900=250

		250	
HEX	FA		
DEC	250		
ост	372	0001 0# 8****	
RIN	1111 1010	CSDN @ 想成采购	即改为FA 00 00 00

ctfshow{dd7d8bc9e5e873eb7da3fa51d92ca4b7}

{there_is_no_flag_here}

CSDN @想成菜鸡

misc25

png格式与bmp不同,他的宽高部分不一样且不需要大端优先也没那么复杂直接改

r⇔ffse⊾	0	1	2	3	- 4	- 5	6	- 7	- 25 9	A	В	C	D	E	F	
10 00000	89	50	4E	47	OD	0A	1A	ΟA	 000	00	OD	49	48	44	52	I
00000010	00	00	03	84	00	00	00	96	08 02	00	00	SP	76	電砂	涩門	i II

详解PNG文件结构 - Angel_Kitty - 博客园 (cnblogs.com) ^(分) https://www.cnblogs.com/ECJTUACM-873284962/p/8986391.htmlflag在图片下面。 改高度 00 00 00 96改成00 00 00 F2就行

misc26

flag在下面 但多下面 需要CRC爆破

对一张正常的图片,通过修改其宽度或者高度隐藏信息,使计算出的CRC校验码与原图的CRC校验码不一致;windows的图片查看器会忽略错误的CRC校验码,因此会显示图片,但此时的图片已经是修改过的, 所以会有显示不全或扭曲等情况,借此可以隐藏信息。

而Linux下的图片查看器不会忽略错误的CRC校验码,因此用Linux打开修改过宽或高的png图片时,会出现打不开的情况

爆破图片修改前的宽和高来匹配CRC校验码,并用正确的宽和高来修复图片

```
import zlib
import struct
filename = 'misc26.png'
with open(filename, 'rb') as f:
   all_b = f.read()
    crc32key = int(all_b[29:33].hex(),16)
   data = bytearray(all_b[12:29])
                       #理论上0xfffffffff,但考虑到屏幕实际/cpu,0x0fff就差不多了
   n = 4095
                              #高和宽一起爆破
   for w in range(n):
       width = bytearray(struct.pack('>i', w)) #q为8字节, i为4字节, h为2字节
       for h in range(n):
           height = bytearray(struct.pack('>i', h))
           for x in range(4):
               data[x+4] = width[x]
               data[x+8] = height[x]
           crc32result = zlib.crc32(data)
           if crc32result == crc32key:
               print("宽为: ",end="")
               print(width)
               print("高为: ",end="")
               print(height)
               exit(0)
```

然后winhex修改 得到



ctfshow{94aef125e087a7ccf2e28e742efd704c}

misc27 flag在图片下面

给了个jpg 所以没法爆破 直接改高 他的和png还不一样 需要查找

右键点击图片 选择属性 打开

National States												
常规安	全	属性修改	详细信息									
修改格式					格式转换							
文件类型:	jpg₿	图片文件(.jp	g)	支持jp	g、png、bmp等	Ē						
调整大小						_						
大小:	34.3	压缩体积										
占用空间:	36.0 KB (36,864 字节) 支持压缩至指定大小											
分辨率:	900	x 150										
宽度:	900	像素			修改尺寸							
高度:	150	像素		支持指	定尺寸、高清质量	Ľ						
图片输出												
名称:	mis	c27.jpg			打印							
位置:	E:\c	tfshow		支持排	版美化、高清品质 CSDN @想成菜鸡的	i 的武阳						

他的高度为150像素 改成16进制为96 宽改后为03 84 winhex搜96 0384

Offset 🔺	搜索约	課													时间]					
9E	96														202	2/04	/27 1				
9F	0384														202	2/04	1/27 1				
165	96														202	2/04	1/27 1				
21A	96														202	2/04	1/27 1				
825	96														202	2/04	1/27 1				
8BB	96														202	2/04	1/27 1				
8C5	96														202	2/04	1/27 1				
96E	96														202	2/04	1/27 1				
AD3	96														202	2/04	/27 1				
BC7	96														202	2/04	/27 1				
BCB	96														202	2/04	1/27 1				
C5E	96														202	2/04	/27 1				
C71	96														202	2/04	1/27 1				
DE9	96														202	2/04	/27 1				
F99	96														202	2/04	4/27 1				
FFF	96														202	2/04	4/27 1				
1062	96														202	2/04	1/27 1				
10ED	96														202	2/04	4/27 1				
1440	96														202	2/04	/27 1				
Offset	0	1	2	3	4	5	6	7	8	9	A	В	C	D	E	F	$\ominus =$			~	
00000000	FF	D8	FF	ΕE	00	0E	41	64	6F	62	65	00	64	40	00	00	ÿØÿî	Adobe	e d@		
00000010	00	01	FF	DB	00	84	00	02	02	02	02	02	02	02	02	02	ÿÛ	I .			
00000020	02	03	02	02	02	03	04	03	02	02	03	04	05	04	04	04					
00000030	04	04	05	06	05	05	05	05	05	05	06	06	07	07	08	07					
00000040	07	06	09	09	ΟA	ΟA	09	09	OC	OC	0C	OC	OC	0C	0C	0C					
00000050	0C	OC	OC	0C	OC	OC	OC	01	03	03	03	05	04	05	09	06					
00000060	06	09	OD	ΟA	09	ΟA	OD	OF	ΟE	ΟE	ΟE	ΟE	OF	OF	OC	OC					
00000070	0C	OC	OC	OF	OF	OC	OC	OC	OC	0C	OC	OF	OC	0C	0C	OC					
00000080	0C	OC	OC																		
00000090	OC	OC	OC	OC	OC	OC	OC	OC	FF	CO	00	11	08	00	<mark>9</mark> 6	03		ÿÀ			
000000A0	84	03	01	11	00	02	11	01	03	11	01	FF	DD	00	04	00	1		ÿΫ́		
00000080	71	FF	C4	01	A2	00	00	00	07	01	01	01	01	01	00	00	qÿÄÇ	SDN @	》想成3	哀观	的武阳
∥ იიიიიი⊂ი	0.0	ΠN	ΠN	ΠN	ΠN	ΠN	Π4	05	03	Π2	Π6	Π1	ΠN	Π7	Π8	Π9					211 2 - 11 I I

总结:

- 1.查看图像属性详细信息是否有隐藏内容或者看exif信息 EXIF信息查看器 (tuchong.com)
- 或者用exiftools
- 2.利用winhex或nodepad++打开搜索ctf,CTF,flag,key等关键字是否存在相关信息
- 搜下有没有txt 有的话 直接扔kali里分解
- 3.检查图像的开头标志和结束标志是否正确,若不正确修改图像标志恢复图像,打开查看是否有flag或ctf信息,(往往gif属于动图,需要分帧查看各帧图像组合所得数据若不是直接的ctf或flag信息需要考虑将其解码)看标志位是否损坏没有的话加 winhex里:右键-》编辑-》粘贴0字节-》插入所需要的位数-》修改那些插入的0字节
- jpg图像开始标志: FF D8 结束标志: FF D9
- gif图像开始标志: 47 49 46 38 39 61 (GIF89)结束标志: 01 01 00 3B
- bmp图片开始标志: 42 4D //92 5B 54 00 00 00 00 00 结束标志: 00
- png图片开始标志: 89 50 结束标志: 60 82
- 4.将图片放置在kail系统中,执行binwalk xxx.jpg 查看图片中是否是多个图像组合或者包含其他文件(若存在多幅图像组合 用binwalk来找 语法: binwalk -e 文件路径,再执行foremost xxx.jpg会自动分离;若检测出其他文件修改其后缀名即可, 如zip)

binwalk -e filename

- 5.使用StegSolve对图像进行分通道扫描,查看是否为LSB隐写
- 6.在kail下切换到F5-steganography,在java Extract运行
- 命令: java Extract 123456.jpg图片的绝对地址 -p 123456
- 判断是否为F5算法隐写
- 7.在kali系统中使用outguess-master工具(需要安装),检测是否为guess算法隐写
- 8.用winhex改变像素
- 其他人的思路:

2.docx文件类型:

- (1) 文档中含有隐藏文字,选项中设置。
- (2) 在kali下改为压缩包,看一下含有的隐藏信息。
- 3.jpg图片文件:
- (1) 查看属性, notepad++打开, 16进制打开看格式, 看看有没有关键字。
- (2) 备份一份,改为zip,看看是否包含其余的文件。
- (3) jpg文件kali查看文件, kali下的命令: binwalk 文件名 分离文件: foremost -e 文件名

0

outguess隐写: outguess -r angrybird.jpg angrybird.txt。

steghide: 查看隐藏在文件中的信息: steghide info 文件名,分离文件: steghide extract -sf 文件名。

- (4) 使用stegsolve, 查看不同通道, 不同偏移量是否含有其余信息。
- (5) jpg图片可以使用stegdetect -tjopi -s 10.0 文件名查看是什么隐藏方式。
- (6) jpg文件, jphide可以使用steghide解密 (jphs),命令: steghide extract -sf 文件名 (要密码)

(7) jpg文件下的F5隐写,进入F5-steganography-james文件夹,在空白处 ctrl+shift+鼠标右键->在此处打 开命令窗口,在cmd中输入命令: java Extract 文件名 -p 密码; kali下也可以使用java Extract /root/文件名 -p 密码提取F5隐写文件,在F5文件夹中可以找到output.txt

(8) jpg文件可能会用到brainftools分离,命令: bftools.exe decode braincopter 文件名 --output out.jpg之 后运行: bftools.exe run out.jpg

4.png文件:

- (1) 查看属性, notepad++打开, 16进制打开看格式, 看看有没有关键字。
- (2) kali查看文件: binwalk 文件名
- 分离文件: foremost -e 文件名

(3) 使用stegsolve,查看不同通道,不同偏移量是否含有其余信息。

(4) brainftools分离,命令: bftools.exe decode braincopter 文件名 --output out.txt,如果有BrainFuck代码可以运行: bftools.exe decode braincopter 文件名 --output out.jpg,之后运行bftools.exe run out.jpg
(5) steganography软件可以提取文件,选择decrypt为密码提取。(web版steganography: http://www.atool.org/steganography.php)

- (6) tweakpng判断是不是png格式,可能校验位有问题。
- (7) 在十六进制编辑器中修改高度(二行六列)查看隐藏信息。
- 5.bmp文件:
- (1) 查看属性, notepad++打开, 16进制打开看格式, 看看有没有关键字。
- (2) kali查看文件: binwalk 文件名
- 分离文件: foremost -e 文件名
- (3) 使用stegsolve,查看不同通道,不同偏移量是否含有其余信息。
- (4) bmp文件隐写可能是LSB,利用Wbstego解决,生成一个is文件,文本编辑器打开看看。
- (5) 把图片放到画图里,改成png格式保存,再利用png的隐写查看隐藏信息。
- 6.gif文件:
- (1) notepad++查看文件头GIF8, 十六进制打开: 修改头文件
- (2) 使用stegsolve, 逐帧查看。
- 7.zip文件:
- (1) 密码爆破
- 8.音频文件:
- (1) 查看属性, notepad++打开, 16进制打开看格式, 看看有没有关键字。
- (2) MP3stego命令: Decode.exe -X-P 密码 文件名
- (3)利用Audacity分析音频文件。
- (4) kali查看文件: binwalk 文件名 分离文件: foremost -e 文件名
- 9.stegsolve
- (1)下面右选项表示不同的色素的通道
- (2) 数据提取, analysis, date extract, 按照选项选完(RGB)
- (3) 图片合成,先打开一张,用中间的analysis,combine,二维码一般四角为黑色,可以进行反色操作(点下面的左右)
- 10.其余文件:
- (1) 查看属性, notepad++打开, 16进制打开看格式, 搜索关键字(flag, ctf, key)。
- (2) 两张以上的文件可能会使用stegsolve进行合成。
- (3) 修改zip, 查看隐藏文件。
- (4) 复杂图片可以分离图层。

WinHex: Hex Editor & Disk Editor, Computer Forensics & Data Recovery Software http://www.winhex.com/winhex/BPG Image format (bellard.org) https://bellard.org/bpg/



赢取流量/现金/CSDN周边激励大奖