MISC总结——隐写术(二)



上一篇详情请见: https://www.cnblogs.com/lxz-1263030049/p/9388511.html

本文参考自: 先知社区: https://xz.aliyun.com/t/1836

隐写术介绍:

隐写术是关于信息隐藏,即不让计划的接收者之外的任何人知道信息的传递事件(而不只是信息的内容)的一门技巧与科学。

英文写作Steganography,而这篇内容将带大家了解一下CTF赛场上常见的图片隐写方式,以及解决方法。有必要强调的是, 隐写术与密码编码是完全不同的概念。

第一部分:基于文件结构的图片隐写

背景知识:

首先这里需要明确一下我这里所说的文件结构是什么意思。文件结构特指的是图片文件的文件结构。

我们这里主要讲的是PNG图片的文件结构。

PNG,图像文件存储格式,其设计目的是试图替代GIF和TIFF文件格式,同时增加一些GIF文件格式所不具备的特性。

是一种位图文件(bitmap file)存储格式,读作"ping"。PNG用来存储灰度图像时,灰度图像的深度可多到16位,存储彩色图像时

,彩色图像的深度可多到48位,并且还可存储多到16位的α通道数据。

对于一个正常的PNG图片来讲,其文件头总是由固定的字节来表示的,以16进制表示即位 89 50 4E 47 0D 0A 1A 0A,这一部 分称作文件头。

标准的PNG文件结构应包括:

- • PNG文件标志
 - PNG数据块
 PNG图片是有两种数据块的,一个是叫关键数据块,另一种是辅助数据块。正常的关键数据块,定义了4种标准数据块,个PNG文件都必须包含它们。
 它们分别是长度,数据块类型码,数据块数据,循环冗余检测即CRC。
 我们这里重点先了解一下,png图片文件头数据块以及png图片IDAT块,这次的隐写也是以这两个地方位基础的。
 png图片文件头数据块
 即IHDR,这是PNG图片的第一个数据块,一张PNG图片仅有一个IHDR数据块,它包含了哪些信息呢?IHDR中,包括了图片的宽,高,图像深度,颜色类型,压缩方法等等。

	Ó	1	2	3	4	5	é	7	8	9	A	B	ç	D	E	F	0123456789ABCDEF
0000h:	89	50	4E	47	0D	0A	1A	0A	þo	00	00	0D	49	48	44	52	%PNGIHDR
0010h:	00	00	02	98	00	00	02	98	80	02	00	00	00	58	53	1F	. <u>~XS.</u>
0020h:	9E	00	01	00	00	49	44	41	54	78	9C	EC	FD	EB	B 3	24	žIDATxœìýë³\$
0030h:	49	72	1F	8A	FD	DC	23	B2	AA	CE	B 3	DF	ЗD	D3	ЗD	CF	Ir.ŠýÜ#°°Î³ß=Ó=Ï
0040h:	9D	9D	D7	EE	62	81	05	76	81	5D	10	24	61	B8	26	13	×îbv.].\$a,&.
0050h:	EC	5E	52	A2	DD	6B	34	7D	90	74	4D	1F	25	93	E9	2F	ì^R¢Ýk4}.tM.%"é/
0060h:	E1	77	C9	F4	99	1F	F4	4D	BA	A2	68	Α4	49	24	C1	17	áwÉô™.ôM°¢h¤I\$Á.
0070h:	08	E0	12	24	C8	DD	05	76	07	3B	33	ЗB	EF	99	EE	9E	.à.\$ÈÝ.v.;3;ï™îž
0080h:	7E	9C	3E	A7	CF	A3	AA	32	C3	5D	1F	3C	22	32	2A	33	~œ>§Ï£ª2Ã].<"2*3
0090h:	AB	FA	9C	D3	A 7	1F	33	8B	98	33	D9	55	59	99	F1	FO	≪úœÓ§.3<~3ÙUY™ñð
00A0h:	FO	FO	9F	BB	47	84	07	FD	9F	7F	F6	FF	20	85	02	8E	ððŸ»G".ýŸ.öÿŽ
00B0h:	19	20	40	15	00	D1	F4	E8	68	32	59	03	B4	69	82	68	. @Ñôèh2Y.´i,h
00C0h:	68	EA	66	6F	6F	EF	DE	BD	3B	93	C9	64	7D	7D	B2	BE	hêfooïÞ%;"Éd}}*%
00D0h:	BE	3E	1A	8D	00	06	C0	EC	98	89	19	22	D2	34	4A	00	¾>Àì~‰."Ò4J.

如图中蓝色的部分即IHDR数据块。

IDAT 数据块

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。这是一个可以存在多个数据块类型的数据块。它的作用就 是存储着图像真正的数据。

因为它是可以存在多个的,所以即使我们写入一个多余的IDAT也不会多大影响肉眼对图片的观察

高度被修改引起的隐写:

背景知识中,我们了解到,图片的高度,宽度的值存放于PNG图片的文件头数据块,那么我们就是可以通过修改PNG图片的 高度值,来对部分信息进行隐藏的。

- 在实验中找到隐写术目录,打开图片隐写,打开图片隐写第二部分文件夹
- 在该文件夹找到 hight.png,
- 双击打开图片,我们先确认一下图片内容并没有什么异常
- 正如前文所说,我们这个实验部分讲的是图片高度值被修改引起的的隐写方式,所以我们010Editor
- 在010Editor运行PNG模板,这样方便于我们修改PNG图片的高度值
- 找到PNG图片高度值对应的地方,然后修改为一个较大的值,并重新计算,修改CRC校验值,并保存文件
- 打开保存后的图片,发现底部看到了之前被隐写的信息

用010Editor打开图片,运行PNG模板

10 editor呢?因为这个16进制编辑器,有模版功能,当我们运行模版后,可以轻易的找到图片的各个数据块的位置以及内容。

Solo Editor - C:\Users\59228\Desktop\hight.png	- 🗆 X											
Startup 💿 hight. png 📓												
0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCPEF												
0000h: 89 50 4E 47 0D 0A 1A 0A 00 00 0D 49 48 44 52 KPNGIHDR												
0020h; 86 00 00 09 70 48 59 73 00 00 12 74 00 00 12 *												
0030h: 74 01 DE 66 1F 78 00 00 0A 4D 69 43 43 50 50 68 t.Pf.xMiCCPPh												
0040h: 6F 74 6F 73 68 6F 70 20 49 43 43 20 70 72 6F 66 otoshop ICC prof												
0050h: 69 6C 65 00 00 78 DA 9D 53 77 58 93 F7 16 3E DF ilexÚ.SwX*÷.>B												
0060h: F7 65 0F 56 42 D8 F0 B1 97 6C 81 00 22 23 AC 08 ÷e.VBØð±-1"‡¬.												
0070h: C8 10 59 A2 10 92 00 61 84 10 12 40 C5 85 88 0A E.Yo.'.a,												
0090h: 50 14 15 11 95 40 55 54 62 D5 04 40 9D 06 22 A0 Vtenda, U.n. a												
00A0h: 7D 7A EF ED ED FB D7 FB BC E7 9C E7 FC CE 79 CF)ziiûwû4cœcûîyî												
00B0h: 0F 80 11 12 26 91 E6 A2 6A 00 39 52 85 3C 3A D8 .€s`#ej.9R<:Ø												
00C0h: 1F 8F 4F 48 C4 C9 BD 80 02 15 48 E0 04 20 10 E6OHĂĖ₩EHàæ												
00D0h: CB C2 67 05 C5 00 00 F0 03 79 78 7E 74 B0 3F FC EAg.A8.yx~t°?ú												
00E0h: 01 AF 6F 00 02 00 70 D5 2E 24 12 C7 E1 FF 83 BA . opo.\$.Çáýf°												
00F001: 50 26 57 00 20 91 00 E0 22 12 E7 0E 01 90 52 00 F6W. '.a".ÇK.												
0110h: 00 00 6C 79 7C 42 22 00 AA 0D 00 FC F4 49 3F 05 10 URL * 10T.	*											
輸出 弓× Inspector	8 ×											
(This panel displays Name Value Start Size Color Comment												
(Press Esc to hide th												
□ 輸出 🔄 查找 🜔 → 🖸 自动 🔄 变量 🖻 书签 🖉 函数	$\langle \cdot \rangle$											
Opened file 'C/\Users\\59228\Desktop\hightpng'.												
	L. L											

10 editor这个16进制编辑器,有模版功能,当我们运行模版后,可以轻易的找到图片的各个数据块的位置以及内容。 找到PNG图片高度值所对应的位置,并修改为一个较大的值

我们找到IHDR数据块,并翻到struct IHDR lhdr位置,修改height的值到一个较大的值,如从700修改到800。 使用CRC Calculator重新计算CRC校验值

📾 CRC Calculator		_		×
Hex ASCII 494844520000021000003200806000000	Сору	Info Name:C	RC-32	
	Paste	Poly: Init: RefIn: RefOut XorOut	0x04C1 0xFFFF True True 0xFFFF	1DB7 FFFF
CRC-32 x32+x26+x23+x22+3 V	alculat	Alias: use:Wi	CRC-32/ nRAR,ec	ADCCI
CRC-32 x32+x26+x23+x22+x16+x12+x11	Copy 1+x10+x8+x7-	+x5+x4+x2	+x+1 Ve	r 0. 🙂

输入参数,然后点击Calculator计算,得到CRC值

为什么要重新计算CRC校验值呢?防止图片被我们修改后,自身的CRC校验报错,导致图片不能正常打开。

修改相应的CRC校验值,为我们重新计算后数值

又(+(F) 溯祖(E) 亘比(5)	1兆四(V) 四平(U)	(俣仮(L) _ 上具(I)	図凵(VV)	₩IJ(E1)						
🖻 - 🙉 - 💷 🖓 📖	🎮 🙈 i 🔍 🗈		0	v R	2			0t		
	📁 🗩 🗄 🐢 🖽			A	P		s ; t = t🛛 t🕹 t	•		
编辑作为:篩舌叚栖 👻 🗚	🔌 🏭 🗐 📑	è 🕪 🗶 🔽 🖋		🍺 🖉 🖡	3	~ 🛛 🎽 🎒	🛚 💹 🌠 🕼 🕅	te.bt 🗸		
Startun hight. png*										
	4 5 6 7 8	9 3 8 C 1		01234567	SABCDEE					
1000b 89 50 4F 47			8 44 52	*PNG	THDR					
0010h: 00 00 02 10	00 00 03 20 08		3 F8 BD	ar norrer	f.@}5					
0020h: BB 00 00 00	09 70 48 59 73	00 00 12 74 0	0 00 12	»pHY	st					
0030h: 74 01 DE 66	1F 78 00 00 0A	4D 69 43 43 5	0 50 68	t.Pf.x	MiCCPPh					
0040h: 6F 74 6F 73	68 6F 70 20 49	43 43 20 70 7	2 6F 66	otoshop	ICC prof					
0050h: 69 6C 65 00	00 78 DA 9D 53	77 58 93 F7 1	6 3E DF	ilexÚ.	SwX"÷.>ß					
0060h: F7 65 0F 56	42 D8 F0 B1 97	6C 81 00 22 2	3 AC 08	÷e.VBØð÷	-1"#					
0070h: C8 10 59 A2	10 92 00 61 84	10 12 40 C5 8	5 88 0A	È.Yo.'.a						
0080h: 56 14 15 11	9C 48 55 C4 82	D5 0A 48 9D 8	8 E2 A0	VœHUÄ	.Õ.н.^â					
0090h: 28 B8 67 41	8A 88 5A 8B 55	5C 38 EE 1F D	C A7 B5	(, gAŠ^Z(U\8î.ܧu					
DOAOh: 7D 7A EF ED	ED FB D7 FB BC	E7 9C E7 FC C	E 79 CF	}zïííû×û	ucœcüÎyÏ					
00B0h: 0F 80 11 12	26 91 E6 A2 6A	00 39 52 85 3	C 3A D8	.€&`æ¢	1.9R<:Ø					
00C0h: 1F 8F 4F 48	C4 C9 BD 80 02	15 48 E0 04 2	0 10 E6	OHÄÉ₩€	Hàæ					
00D0h: CB C2 67 05	C5 00 00 F0 03	79 78 7E 74 E	0 3F FC	ËÂg.Åð	.vx~t°?ü					
DOE0h: 01 AF 6F 00	02 00 70 D5 2E	24 12 C7 E1 F	F 83 BA	opõ	.\$.Cáÿf°					
DOFOh: 50 26 57 00	20 91 00 E0 22	12 E7 0B 01 9	0 52 00	P&Wà	".çR.					
0100h: C8 2E 54 C8	14 00 C8 18 00	B0 53 B3 64 0	A 00 94	È.TÈÈ.	.°S'd"					
0110h: 00 00 6C 79	7C 42 22 00 AA	OD 00 EC F4 4	9 3E 05	ly B".	ªìôI>.					
0120h: 00 D8 A9 93	DC 17 00 D8 A2	1C A9 08 00 8	D 01 00	.ø©"Üø	¢.©					
0130h: 99 28 47 24	02 40 BB 00 60	55 81 52 2C 0	2 CO C2	™(G\$.@».	`U.R,.ÀÂ					
0140h: 00 A0 AC 40	22 2E 04 CO AE	01 80 59 B6 3	2 47 02	@"À	©.€Y¶2G.					
0150h: 80 BD 05 00	76 8E 58 90 OF	40 60 00 80 9	9 42 2C	€%vŽX.	.@`.€™B,					
11 COL . CC 00 20 20	02 00 42 15 12	CD 02 20 4C 0	0 20 00	t o c	ŕτ ο					
Inspector - PNGTemplate.bt	:									
Name	Value	Start	Size		Color	Comment				
uint64 pagid	89504E470D0A1	Ob	8h	For	Bar					
✓ struct CHUNK chunk[0]	IHDR (Critical R.,	8h	19h	Fa:	Ba:					
uint32 length	13	8h	4h	Fg:	Bg:					
> union CTYPE type	IHDR	Ch	4h	Fg:	Bg:					
> struct IHDR ihdr	528 x 800 (x8)	10h	Dh	Fg:	Bg:					
uint32 crc	A3F8BDBBh	1Dh	4h	Fg:	Bg:					
> struct CHUNK chunk[1]	iCCP (Ancillary,	21n 26h	15h 450b	Fg:	Bg: Bg:					
struct CHUNK chunk[2]	cHRM (Ancillary	A8Fh	2Ch	For	Bg:					
> struct CHUNK chunk[4]	IDAT (Critical, P.,	ABBh	18F592h	Fa:	Bg:					
> struct CHUNK chunk[5]	IEND (Critical, R.,	19004Dh	Ch	Fg:	Bg:					
•						111				

各位小伙伴可以思考一下: JPG图片是否也有这样的隐写形式呢?

如果感兴趣可以:了解JPG以及GIF等图片文件的格式。

第二部分:隐写信息以IDAT块加入图片

在背景知识中,我们提到了一个重要的概念就是图片的IDAT块是可以存在多个的,这导致了我们可以将隐写西信息以IDAT块 的形似加入图片

- 在实验机找到隐写术目录,打开图片隐写,打开图片隐写第二部分文件夹
- 在该文件夹找到 hidden.png,
- 双击打开图片,我们先确认一下图片内容并没有什么异常
- 使用pngcheck先对图片检测
- 在pngcheck的检测下,我们会发现异常信息,我们对异常的块进行提取
- 编写脚本,提取异常信息(关于脚本,你可以上网搜索,也可以自己写)



我们使用命令:

pngcheck -v hidden.png

对图片的文件结构进行检测。

发现异常,并判断异常的原因 我们会发现,图片的的数据块形式是如下的 Type: IHDR Size: 13 CRC: 5412913F

Pos: 33 Type: IDAT Size: 10980 CRC: 98F96EEB

Pos: 11025 Type: IEND Size: 0 CRC: AE426082

我们会惊讶的发现pos为11025的size居然为0,这是一块有问题的地方,我们可以怀疑,这一块是隐写的信息。

自己编写脚本代码(也可以去网上搜索):

```
#!/usr/bin/python
```

```
from struct import unpack
from binascii import hexlify, unhexlify
import sys, zlib
# Returns [Position, Chunk Size, Chunk Type, Chunk Data, Chunk CRC]
def getChunk(buf, pos):
    a = []
    a.append(pos)
    size = unpack('!I', buf[pos:pos+4])[0]
    # Chunk Size
```

```
a.append(buf[pos:pos+4])
    # Chunk Type
    a.append(buf[pos+4:pos+8])
    # Chunk Data
    a.append(buf[pos+8:pos+8+size])
    # Chunk CRC
    a.append(buf[pos+8+size:pos+12+size])
    return a
def printChunk(buf, pos):
    print 'Pos : '+str(pos)+''
    print 'Type: ' + str(buf[pos+4:pos+8])
    size = unpack('!I', buf[pos:pos+4])[0]
    print 'Size: ' + str(size)
    #print 'Cont: ' + str(hexlify(buf[pos+8:pos+8+size]))
    print 'CRC : ' + str(hexlify(buf[pos+size+8:pos+size+12]).upper())
    print
if len(sys.argv)!=2:
    print 'Usage: ./this Stegano_PNG'
    sys.exit(2)
buf = open(sys.argv[1]).read()
pos=0
print "PNG Signature: " + str(unpack('ccccccc', buf[pos:pos+8]))
pos+=8
chunks = []
for i in range(3):
    chunks.append(getChunk(buf, pos))
    printChunk(buf, pos)
    pos+=unpack('!I',chunks[i][1])[0]+12
decompressed = zlib.decompress(chunks[1][3])
# Decompressed data length = height x (width * 3 + 1)
print "Data length in PNG file : ", len(chunks[1][3])
print "Decompressed data length: ", len(decompressed)
height = unpack('!I',(chunks[0][3][4:8]))[0]
width = unpack('!I',(chunks[0][3][:4]))[0]
blocksize = width * 3 + 1
filterbits = ''
for i in range(0,len(decompressed),blocksize):
    bit = unpack('2401c', decompressed[i:i+blocksize])[0]
    if bit == '\x00': filterbits+='0'
    elif bit == '\x01': filterbits+='1'
    else:
        print 'Bit is not 0 or 1... Default is 0 - MAGIC!'
        sys.exit(3)
s = filterbits
endianess_filterbits = [filterbits[i:i+8][::-1] for i in xrange(0, len(filterbits), 8)]
flag = ''
for x in endianess filterbits:
```

```
if x=='00000000': break
flag += unhexlify('%x' % int('0b'+str(x), 2))
```

```
print 'Flag: ' + flag
```

最后得到答案: flag DrgnS{WhenYouGazeIntoThePNGThePNGAlsoGazezIntoYou}

■题目.zip (0.414 MB) 下载附件

您可以考虑给博主来个小小的打赏以资鼓励,您的肯定将是我最大的动力。



作者: 落花四月

出处: https://www.cnblogs.com/lxz-1263030049/

关于作者: 潜心于网络安全学习。如有问题或建议, 请多多赐教!

版权声明:本文版权归作者和博客园共有,欢迎转载,但未经作者同意必须保留此段声明,且在文章页面明显位置给出原文连接.

特此声明:所有评论和私信都会在第一时间回复。也欢迎园子的大大们指正错误,共同进步。或者直接私信我 声援博主:如果您觉得文章对您有帮助,可以点击文章右下角【推荐】一下。您的鼓励是作者坚持原创和持续写作的最大动 力!



<u>创作打卡挑战赛</u> 赢取流量/现金/CSDN周边激励大奖