CTF show STEGA系列

原创

yu22x ● 于 2020-02-19 21:01:24 发布 ● 2661 ♥ 收藏 6 分类专栏: <u>CTF show STEGA系列</u>文章标签: <u>python linux 安全 密码学</u> 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/miuzzx/article/details/104397527</u> 版权



CTF show STEGA系列 专栏收录该内容

1篇文章1订阅 订阅专栏

题目地址: https://ctf.show

一般我做图片隐写类的题目有以下几个步骤:
1.如果是在压缩包中的图片,首先查看压缩包的二进制,没有异常,解压压缩包。
2.查看图片的属性,是否有附加的信息。
3.尝试打开图片,不能打开可能文件头缺少或其他原因。能打开,查看图片二进制。
4.在二进制中查看是否有其他类型的文件头。有则分离出来。
4.放入stegsolve中查看是否是LSB隐写。
5.最后实在没有办法就隐写工具挨个尝试吧

0x01 stega1

这道题就不啰嗦了,使用隐写工具jphs 链接: https://pan.baidu.com/s/1Oq8Wektf-JQSmJOwSYb8BA 提取码: 4qks

题目中没有任何提示,直接尝试空密码解密,成功获取flag。

0x02 stega2

一般图片的宽高被修改过,放到linux中是打不开的,对于这张图,可以直接爆破宽高,附上脚本:

	Ó	1	2	3	4	5	6	7	8	9	A	B	Ċ	D	E	F	0123456789ABCDEF
h:	89	50)	4E	47	0D	0 A	1 A	0A	00			0D	49	48	44	52	₽NG IHDR
h:	00		02	28			01	AO		02				3D	9A	65	(=š
h:	DU	00	00	20	00	49	.7.7	-1	54	78	01	EC	BD	59	73	24	∋IDATx.ì½Ys\$
h:	49	92	E7	87	23	70	03	09	20	8F	CA	AC	в3	EB	EA	AE	I′ç‡#pʬ³ëê®
h:	9E	D9	D9	DE	95	E5	7E	55	3E	F3	7в	2C	\mathbf{DF}	F 8	в8	4 F	žÙÙ₽∙å~U>ó{,ßø,O

所以对于上面的脚本,不同的图片只要修改crc32的值即可。

0x03 stega3

解压得到一张图片,010editor查看,16进制的末尾有提示是ntfs隐写(前提必须用winrar解压),所以直接上工具(链接:https://pan.baidu.com/s/1Nw70ASUv-sgurDZnFDte4w提取码:zbw4)得到flag.txt文档,打开得flag。

0x04 stega4

两张相同的图片,首先尝试放入stegsolve中,在stegsolve中有图片混合(image combiner)的功能如下图,尝试无果,采用盲 水印解密。这里附上工具(链接: https://pan.baidu.com/s/1BoWao6bJ5wvmQqqYJtF-ew 提取码: 3tba),对于python脚本,提 取图片中的盲水印: python bwm.py decode 0.png 1.png 2.png 其中0.png和1.png为得到得两张相同的图片,2.png为解密生 成的图片。得到结果如下:



0x05 stega10

在图片的二进制中获取了一串疑似base64的字符串如下,尝试解码,失败,观察到字符串后面还有两个空位,果断添加上两个 等于号,成功解密base64,获得一个下载地址。

			<u> </u>		<u> </u>							<u> </u>		<u></u>	<u>.</u>	0110100100100000
FF	D8	$\mathbf{F}\mathbf{F}$	E0		10	4A	46	49	46	00	01	01			01	ÿØÿàJFIF
00	01		00	FF	DB	00	43	00	80	06	06	07	06	05	08	ÿÛ.C
07	07	07	09	09	08	0A	0C	14	0D	0C	0B	0в	0C	19	12	
13	0F	14	1D	1A	1F	1E	1D	1A	1C	1C	20	24	2E	27	20	\$.'
22	2C	23	1C	1C	28	37	(29)	2C	30	31	34	34	34	1F	27	", # (7)),01444.'
39	ЗD	38	32	3C	2E	33	34	32	FF	DB	00	43	01	09	09	9=82<.342ÿÛ
09	0C	0B	0C	18	0D	0D	18	32	21	1C	21	61	48	52	30	2'.!aHR0
63	48	4D	36	4C	79	39	33	64	33	63	75	62	47	46	75	cHM6Ly93d3cubGFu
65	6D	39	31	63	79	35	6A	62	32	30	76	61	54	6C	69	em91cy5jb20vaTli
4D	47	74	7A	5A	41	00	00	32	32	32	32	32	32	32	32	GtzZA. 2222222
32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	222222222222222222222222222222222222222
32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	2222222222222222222
32	32	32	32	32	32	32	32	32	FF	C0	00	11	08	01	۱ЩQS:/	nordersaithenunzzy

下载下来以后有一个密码文件夹,一个加密的压缩包。打开密码文件夹,我们发现每一个文档都是一字节,这时我们可以根据文档的crc32爆破里面的值,运行如下脚本得到密码为447^*5#)7



我在这里再多说一句话,万一我们碰上文件夹中有很多txt怎么办呢,不可能一个一个自己输入吧,我再附上一个比较实用的脚本:



我们用得到的密码解压加密的压缩包,得到一张图片,但是无法打开,用010editor打开,发现是将图片的16进制逆序了,修复脚本如下:

```
f = open("n.png","rb")
f1 = open("m.png","wb")
s = f.read()
s= s[::-1]
f1.write(s)
f.close()
f1.close()
```

得到一张二维码, 扫码得到flag

0x06 stega11

我们直接用010editor打开,发现图片中还包含一张图片(字节搜索ffd8),更有趣的是两个图片之间隐藏了一段base32,直接 解密得flag

04	44	40	44	44	04	44	40	44	44	04	44	40	44	44	04	.D@DD.D@DD.D@DD.
44	40	44	44	04	44	40	44	44	04	44	40	44	44	04	44	Dependen napp.p
40	44	44	1F	$\mathbf{F}\mathbf{F}$	D9	4D	5A	57	47	43	5A	33	33	47	5A	@DD.ÿÙMZWGCZ33G2
54	44	43	4E	5A	5A	47	35	53	44	49	4D	42	59	47	42	TDCNZZG5SDIMBYGB
52	44	45	4F	4C	43	47	59	32	47	49	59	4A	56	48	41	RDEOLCGY2GIYJVHA
34	54	4F	4E	5A	59	47	41	32	44	4D	4D	33	46	47	4D	4TONZYGA2DMM3F
59	48	32	FF	D8	FF	E0	00	10	4A	46	49	46	00	01	01	1 2 #ØÜ>E
01	00	48	00	48	00	00	FF	DB	00	43	00	06	04	05	06	<u>н.ну</u> б.с