Bugku-杂项部分题目WP



<u>晚安這個未知的世界</u> U 于 2020-10-11 12:50:01 发布 O 1530 🏠 收藏 3 分类专栏: Bugku 杂项 文章标签: 信息安全 加密解密 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: https://blog.csdn.net/weixin 47598409/article/details/109010457 版权 Bugku 杂项 专栏收录该内容

1篇文章0订阅 订阅专栏

前言

一个新手学习CTFer的成长之路必定要通过大量的刷题,下面是bugku部分题目wp

Bugku-MISC-论剑



十年磨一剑,霜刃未曾试。 今日把示君, 谁有不平事。

lunjian.jpg

Flag

Submit

起始	迈	1.	mji	an.	jpg	×												
¥	编辑	方式	: +:	六进'	制(H) 🗸	jź	行脚	本、	<pre>/</pre>	运行	模板	~				1000	
		Ŏ					5						B	Ç	D	E		0123456789ABCDEF
248	Oh:	7E	FO	6E	84	AE	9F	F9	C5	6D	50	79	03	71	F9	90	EO	~ðn"ØŸùÅmPy.qù.à
249	0h:	7C	CD	3F	DB	29	48	20	2A	87	CE	AB	F5	88	00	65	E3	ĺ?Û)H *‡Î«õ^.eã
24A	0h:	12	2D	AC	30	41	27	82	2E	F9	9F	19	E2	2F	DE	4E	F6	0A',.ùŸ.â/ÞNö
24B	0h:	9F	BF	E3	13	52	A8	94	40	92	El	07	6F	7F	54	58	20	Ÿ¿ã.R [~] "@'á.o.TX
24C	0h:	01	Al	B3	9E	84	E2	1E	79	C8	04	14	01	D8	D9	C7	66	.; ³ ž"â.yÈØÙÇf
24D	Oh:	F3	4F	10	E2	9E	30	B4	5F	70	BF	8F	FD	ЗA	FF	D9	30	óO.âž0′ p¿.ý:ÿĎ0
24E)	Oh:	31	31	30	31	31	30	31	20	30	31	31	31	31	30	30	31	1101101 01111001
24F)	0h:	20	30	31	31	30	31	31	31	30	20	30	31	31	30	30	30	01101110 011000
250	0h:	30	31	20	30	31	31	30	31	31	30	31	20	30	31	31	30	01 01101101 0110
251	Oh:	30	31	30	31	20	30	31	31	30	31	30	30	31	20	30	31	0101 01101001 01
252	0h:	31	31	30	30	31	31	20	30	31	31	30	31	30	31	31	20	110011 01101011
253	Oh:	30	31	31	30	30	31	30	31	20	30	31	31	31	31	30	30	01100101 0111100
254	0h:	31	20	30	30	31	30	30	30	30	31	20	30	30	31	30	30	1 00100001 00100
255	Oh:	30	30	31	20	30	30	31	30	30	30	30	31	20	30	31	31	001 00100001 011
256	0h:	30	31	30	30	30	20	30	31	31	30	31	30	30	30	20	30	01000 01101000 0
257	Oh:	31	31	30	31	30	30	30	38	7B	BC	AF	27	1C	00	04	CB	11010008{¥, 'Ë
258	0h:	B2	17	DF	A0	1E	00	00	00	00	00	00	6A	00	00	00	00	*.18]
259	Oh:	00	00	00	6D	C5	15	lF	D8	C3	E1	A 9	0E	D6	5B	F2	3http:	s://blog.csdn.n@i/s@ixini_c47598409
253	016 +	02	55	70	72	52	DO	07	DE	10	5D	64	FC	62	PO	02	60	

1	011011	101	01111001	01101110	01100001	01101101	01100101	01101001	01110011	01101011	01100101	01111001
	001000	001	00100001	00100001	01101000	01101000	01101000					
-												
1	清空	13	交换位置	示例	转换	保存结果	口复制结	果				
1	myname	eisk	ey!!!hhh									

把图片放进010发现有一串二进制先转ASCII码获得一个关键词,但不知道是什么东西

然后通过搜索FFC2来修改JPEG图片的高度,把00改为01

Offset A	損	訪	結果															时间				
	9E FF	C2																2020/09	9/08	23:36:		
45	53F FF	C2																2020/09	9/08	23:36:		
Offset	Ť	0	1	2	3	4	5	6	7	8	9	A	в	С	D	E	F	A=	ANS	SI ASCII	^	
0000003	0	13	OF	14	1D	1A	1F	1E	1D	1A	10	10	20	24	2E	27	20	Ψ=-		\$.'		lunjian.j
0000004	0 :	22	2C	23	10	10	28	37	29	2C	30	31	34	34	34	1F	27	",#	(7),	01444 '		C:\User

文件大小	9=82<.342ÿÛ C	09	09	01	43	00	DB	FF	32	34	33	2E	3C	32	38	3D	39	00000050
	2! !2222	32	32	32	32	21	10	21	32	18	OD	OD	18	OC	0B	OC	09	00000060
	22222222222222222	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	00000070
缺省编辑	22222222222222222	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	08000000
状态:	22222222222222 <mark>ÿÂ</mark>	C2	FF	32	32	32	32	32	32	32	32	32	32	32	32	32	32	00000090
撤消级数	в "	11	03	01	11	02	00	22	01	03	OD	03	42	01	08	11	00	000000A0
反向撤消	ŸÄ	00	00	00	00	01	01	03	02	00	01	00	1B	00	C4	FF	01	000000B0
1001 Manua	ÿÄ	C4	FF	07	01	06	03	02	05	04	00	00	00	00	00	00	00	00000000
创建时间		00	00	00	00	00	00	00	00	00	00	01	01	01	01	16	00	00000D0
	tps://bloggosdn.net/weixin_	03	10	02	00	01	03	0C	00	DA	FF	02	01	00	00	00	00	000000E0

兼CTF_论剑场

Not flag {666C61677B6D795F6E61 65 4 121 7 hhhh

https://blog.csdn.net/weixin_47598409

L 🖌 🖂 🤇					5	Ш.	5	G	EB 0	10	Ĩ		HE	24	B HEX		\rightarrow + \forall = \Rightarrow \Rightarrow	i (ini) 🖬
lunjian.jpg																		
Offset	0	1	2	3	4	5	6	7	8	9	A	В	С	D	E	F	ANSI ASCII	~
00002460	53	48	19	9A	77	F3	D6	AF	92	EO	47	49	BD	30	40	34	SH šwóC 'àGIH0@4	
00002470	90	B 8	CE	3D	0B	B2	16	93	78	D6	CF	BC	D3	F6	CE	9F	œ,Î= ° °xÖÏ4ÓöΟ	_
00002480	7E	FO	6E	84	AE	9F	F9	C5	6D	50	79	03	71	F9	90	EO	~ðn"&ŸùÅmPy qù à	
00002490	7C	CD	3F	DB	29	48	20	2A	87	CE	AB	F5	88	00	65	E3	ĺ?Û)H *‡Î«õ^ eã	
000024A0	12	2D	AC	30	41	27	82	2E	F9	9F	19	E2	2F	DE	4E	F6	OA', .ùŸ â/ÞNö	-
000024B0	9F	BF	E3	13	52	A 8	94	40	92	El	07	6F	7F	54	58	20	Ÿ¿ã R″″@'á o TX	
000024C0	01	Al	B 3	9E	84	E2	1E	79	C8	04	14	01	D8	D9	C7	66	;'ž"â yÈ ØÙÇf	
000024D0	F3	4F	10	E2	9E	30	B4	5F	70	BF	8F	FD	3A	FF	D9	30	óC âžO′_p; ý:ÿÙO	
000024E0	31	31	30	31	31	30	31	20	30	31	31	31	31	30	30	31	1101101 01111001	
000024F0	20	30	31	31	30	31	31	31	30	20	30	31	31	30	30	30	01101110 011000	
00002500	30	31	20	30	31	31	30	31	31	30	31	20	30	31	31	30	01 01101101 0110	
00002510	30	31	30	31	20	30	31	31	30	31	30	30	31	20	30	31	0101 01101001 01	
00002520	31	31	30	30	31	31	20	30	31	31	30	31	30	31	31	20	110011 01101011	
00002530	30	31	31	30	30	31	30	31	20	30	31	31	31	31	30	30	01100101 0111100	
00002540	31	20	30	30	31	30	30	30	30	31	20	30	30	31	30	30	1 00100001 00100	
00002550	30	30	31	20	30	30	31	30	30	30	30	31	20	30	31	31	001 00100001 011	
00002560	30	31	30	30	30	20	30	31	31	30	31	30	30	30	20	30	01000 01101000 0	_
00002570	31	31	30	31	30	30	30	38	7B	BC	AF	27	1C	00	04	CB	11010008{ ¹ , Ë	
00002580	B2	17	DF	A0	1E	00	00	00	00	00	00	6A	00	00	00	00	° ß j	
00002590	00	00	00	6D	C5	15	1F	D8	C3	El	A9	0E	D6	5B	F2	33	mĂ ØÃá© Ö[ò3	
000025A0	D2	EE	70	A2	52	D9	9F	B 5	1B	5D	C4	EC	63	B 9	0.3	69	Ôîl¢RÛŸu lÄìcª i	

这里猜出应该是一个7z压缩包文件头,应该被出题人改了文件头 7z的文件头是377ABCAF271C,然后修改为正确头部后并且修改文件后缀名为7z



发现了有一个图片,但是这个压缩包被加密了,但是回想一下,开始的时候用二进制转ASCII码有一串东西,猜一猜是不是密码?

好家伙,果然是密码

lic2 inc.



但是这张图片跟刚才的那个也是图片高度被修改了,继续用winhex或者010修改红色框的00改为01,保存

ffset 9E	搜索 FFC2	结果	•														时间 2020/0	9/09 (00:01	5:		
Offset	0	1	2	3	4	5	6	7	8	9	A	В	С	D	E	F	<u>م</u> =	ANS	SI A	ASCII	^	
Offset	0 FF	1 D8	2 FF	3 E0	4	5 10	6 4A	7 46	8	9 46	A 00	B 01	C 01	D 01	E 00	F 60	¢≣ ÿØÿà	ANS	SI A	ASCII	^	
Offset 0000000 0000010	0 FF 00	1 D8 60	2 FF 00	3 E0 00	4 00 FF	5 10 DB	6 4A 00	7 46 43	8 49 00	9 46 08	A 00 06	B 01 06	C 01 07	D 01 06	E 00 05	F 60 08	∲≣ ÿØÿà	ANS JFI ÿÛ C	SI A	ASCII	^	
Offset 0000000 0000010 0000020	0 FF 00 07	1 D8 60 07	2 FF 00 07	3 E0 00	4 00 FF 09	5 10 DB 08	6 4A 00 0A	7 46 43 0C	8 49 00 14	9 46 08 0D	A 00 06 0C	B 01 06 0B	C 01 07 0B	D 01 06 0C	E 00 05 19	F 60 08 12	¢≣ ÿØÿà	ANS JFI ÿÛ C	SI A	ASCII	^	
Offset 0000000 0000010 0000020 0000030	0 FF 00 07 13	1 D8 60 07 0F	2 FF 00 07 14	3 E0 00 09 1D	4 00 FF 09 1A	5 10 DB 08 1F	6 4A 00 0A 1E	7 46 43 0C 1D	8 49 00 14 1A	9 46 08 0D 1C	A 00 06 0C 1C	B 01 06 0B 20	C 01 07 0B 24	D 01 06 0C 2E	E 00 05 19 27	F 60 08 12 20	¢≣ ÿØÿà	ANS JFI ÿÛ C	SI A	\$.'	^	
Offset 0000000 0000010 0000020 0000030 0000040	0 FF 00 07 13 22	1 D8 60 07 0F 2C	2 FF 00 07 14 23	3 E0 00 10 1C	4 00 FF 09 1A 1C	5 10 DB 08 1F 28	6 4A 00 0A 1E 37	7 46 43 0C 1D 29	8 49 00 14 1A 2C	9 46 08 0D 1C 30	A 00 06 0C 1C 31	B 01 06 0B 20 34	C 01 07 0B 24 34	D 01 06 0C 2E 34	E 00 05 19 27 1F	F 60 08 12 20 27	∲ ≣ ÿØÿà ", #	ANS JFI ÿÛ C (7),	014	\$.' 144 '	^	
Offset 0000000 0000010 0000020 0000030 0000040 0000050	0 FF 00 07 13 22 39	1 D8 60 07 0F 2C 3D	2 FF 00 07 14 23 38	3 E0 00 10 1C 32	4 00 FF 09 1A 1C 3C	5 10 DB 08 1F 28 2E	6 4A 00 0A 1E 37 33	7 46 43 0C 1D 29 34	8 49 00 14 1A 2C 32	9 46 08 0D 1C 30 FF	A 00 06 0C 1C 31 DB	B 01 06 20 34 00	C 01 07 08 24 34 43	D 01 06 0C 2E 34 01	E 00 05 19 27 1F 09	F 60 08 12 20 27 09	↓ ÿØÿà ", # 9=82.	ANS JFI ÿÛ C (7), <.342	014 290	\$.' \$44 ' C	^	
Offset 0000000 0000010 0000020 0000030 0000040 0000050 0000060	0 FF 00 07 13 22 39 09	1 D8 60 07 0F 2C 3D 0C	2 FF 00 07 14 23 38 0B	3 E0 00 9 1D 1C 32 0C	4 00 FF 09 1A 1C 3C 18	5 10 DB 08 1F 28 2E 0D	6 4A 00 0A 1E 37 33 0D	7 46 43 0C 1D 29 34 18	8 49 00 14 1A 2C 32 32	9 46 08 0D 1C 30 FF 21	A 00 06 0C 1C 31 DB 1C	B 01 06 20 34 00 21	C 01 07 0B 24 34 43 32	D 01 06 0C 2E 34 01 32	E 00 05 19 27 1F 09 32	F 60 08 12 20 27 09 32	∲ ≣ ÿØÿà ", # 9=82	ANS JFI ÿÛ C (7), <.342 2	014 290 2! !	\$.' \$.' 144 ' C	^	
Offset 0000000 0000010 0000020 0000030 0000040 0000050 0000060 0000070	0 FF 00 07 13 22 39 09 32	1 D8 60 07 0F 2C 3D 0C 32	2 FF 00 07 14 23 38 0B 32	3 E0 00 1D 1C 32 0C 32	4 00 FF 09 1A 1C 3C 18 32	5 10 DB 08 1F 28 2E 0D 32	6 4A 00 0A 1E 37 33 0D 32	7 46 43 0C 1D 29 34 18 32	8 49 00 14 1A 2C 32 32 32	9 46 08 0D 1C 30 FF 21 32	A 00 06 0C 1C 31 DB 1C 32	B 01 06 08 20 34 00 21 32	C 01 07 08 24 34 32 32	D 01 06 0C 2E 34 01 32 32	E 00 05 19 27 1F 09 32 32	F 60 08 12 20 27 09 32 32		ANS JFI ÿÛ C (7), <.342 222222	014 290 21 1	\$.' 444 ' 22222 22222	^	

00000090	32	32	32	32	32	32	32	32	32	32	32	32	32	32	FF	C2	22222222222222 <mark>ÿÂ</mark>	100
000000A0	00	11	08	01	3E	02	6E	03	01	22	00	02	11	01	03	11	> n "	()
000000B0	01	FF	C4	00	1B	00	01	00	02	03	01	01	00	00	00	00	Ϋ́Α	
00000000	00	00	00	00	00	00	00	03	04	02	05	06	01	07	FF	C4	ŸÄ	最
00000D0	00	16	01	01	01	01	00	00	00	00	00	00	00	00	00	00	//blog.csdn.net/weixin_475984	09
000000E0	00	00	00	00	01	02	FF	DA	00	00	0.3	01	00	02	10	0.3	ΰÚ	

棄CTF_论剑场



然后把两张这样的图片的flag数据合在一起 flag{666C61677B6D795F6E616D655F482121487D} 然后去提交,发现不正确,再仔细看了一下flag里面的是十六进制的数值,我们去转换一下ASCII码

16进制到ASCII字符串



flag就这样出来了

flag{my_name_H!!H}

Bugku-MISC-图穷匕见

用winhex打开,结果发现下面有一大串16进制数据,FF D9是JPG文件尾

Offset	0	1	2	3	4	5	6	7	8	9	A	В	C	D	E	F	ANSI ASCII
000052D0	06	A2	A7	2F	54	21	11	68	42	15	02	10	85	00	84	21	¢§/T! hB "!
000052E0	00	84	21	50	24	84	28	1A	10	84	02	10	85	40	84	21	"!P\$"("@"!
000052F0	00	84	21	07	FF	D9	32	38	33	37	32	63	33	37	32	39	": ÿÙ20072c0729
00005300	30	61	32	38	33	37	32	63	33	38	32	39	30	61	32	38	0a28372c38290a28
00005310	33	37	32	63	33	39	32	39	30	61	32	38	33	37	32	63	372c39290a28372c
00005320	33	31	33	30	32	39	30	61	32	38	33	37	32	63	33	31	3130290a28372c31
00005330	33	31	32	39	30	61	32	38	33	37	32	63	33	31	33	32	31290a28372c3132
00005340	32	39	30	61	32	38	33	37	32	63	33	31	33	33	32	39	290a28372c313329
00005350	30	61	32	38	33	37	32	63	33	31	33	34	32	39	30	61	0a28372c3134290a
00005360	32	38	33	37	32	63	33	31	33	35	32	39	30	61	32	38	28372c3135290a28
00005370	33	37	32	63	33	31	33	36	32	39	30	61	32	38	33	37	372c3136290a2837
00005380	32	63	33	31	33	37	32	39	30	61	32	38	33	37	32	63	2c3137290a28372c
00005390	33	31	33	38	32	39	30	61	32	38	33	37	32	63	33	31	3138290a28372c31
000053A0	33	39	32	39	30	61	32	38	33	37	32	63	33	32	33	30	39290a28372c3230
000053B0	32	39	30	61	32	38	33	37	32	63	33	32	33	31	32	39	290a28372c323129
000053C0	30	61	32	38	33	37	32	63	33	32	33	32	32	39	30	61	Ja283/2C3232290a

然后复制这些数据

用Notepad++打开,并用这个软件里面的一个插件进行转换 HEX—》ASCI码

🛃 *C:\	\Users\Ad	ministrato	or\Deskto	p\新建文本	文档 (2).	txt - Note	epad++ [/	Administ	rator]					
文件(F)	编辑(E)	搜索(S)	视图(V)	编码(N)	语言(L)	设置(T)	工具(0)	宏(M)	运行(R)	插件(P)	窗口(W)	?		
6		6 6	1 4 1	1	C a	₽ <mark>₽</mark> ₽ @	ج 🖪		5 ¶]	E 🧔 📡	<u>}</u> 🗈 🧟) 🕪 🔤	
🔚 29. z	zlib 🔀 🔚	README. mo	IX 🔚 o.	txt 🔀 🔚	read. php	🗶 🔚 up]	load php] 🔚 cla	asses, php	🛛 🔚 in	dex. php 🗷	🗎 新建文本文	【档 (2).txt区	
1	(7,7)				- C									
2	(7,8)													
3	(7,9)													
4	(7, 10)													
5	(7,11)													
6	(7, 12)													
7	(7, 13)													
8	(7, 14)													
9	(7, 15)													
10	(7,16)													
11	(7 17)													

看上去有点像坐标点

再看一下图片属性信息





会画图吗???

🖹 *C:\	Us	sers\Administrator\[
文件(F)	4	編辑(E) 搜索(S) 视
ි 🚽	1] 🖻 🗟 🐚 🖨 .
🔡 29. z	lił	b 🔀 🔚 README. m d 🗙
1	7	7
2	7	8
3	7	9
4	7	10
5	7	11
6	7	12
7	7	13
8	7	14
9	7	15
10 hill	7	log.codn.net/weixin_47598409

Linux系统里面有一个专门画图工具,根据坐标数值画图 —— gnuplot



扫描二维码得出flag flag{40fc0a979f759c8892f4dc045e28b820}

Bugku-MISC-Convert



[root@root~20:58:02~Tony]

0×526172211a0700cf90730000d00000000000000f7c07420902c000d09000059220000023e6370190a59 20000006b65792e6a706700f014a319101d911508917c9015eacb02ecd08228a0b868b4d082a05db62880 a0a0da68b3426cb02cc44d29a1445404474882209a501d06c5411514313101044d5dc6c050b5440c14b410 0f71eb11df1ccef7c33c4fb91343c999faa7ee4c473c262bf989d1f54ccd4e88fa3f7ea9f0a88a8a8a8889a 63fec0d2adabad2b8036580df85e031fd3a5ac567c6f6dcc2b8b80d783281200f0cde031c9430502974643 050f80202966e146372501f14fb13a3e058b645bf47c0b162d91a06fc683c257b22568f8162c5b23eba3e0 407c03704f4b03c01ec0b806502b0090374599029f99c6854dc72570c946017729ff72d881bc03efd8cab0 4f5d0a7e63e0e4afa170b9045dcc2e794424fe3e30072ab02b2a312eb0278a20d692e0ead84053d6c52860 8d696d6052daf2b7d07a4c1bae4814d6d8da3c8f05e453b111fc16f79D462D78a20d692e0ead84053d6c52860

把这些十六进制转换为ASCII码



可以看出这是rar文件

复制下来,用010以十六进制导入,并另存为以rar后缀的压缩包,解压获得一张图片

	1 <u> </u>		1					5	1.9	•		r.	ົ	3	1	-	t B 👌	🖻 🔶 🗛 📐 Hex 💱 '
	起始页		.	ar >	۲.													
10	▼ 编辑	方式	: +:	六进	制(H) 3) \	<u>ن</u> ح	行脚	本、	′ <u>,</u> :	运行	模板	×	č	р	न	न	01234567892800FF
mmmrar	0000h:	52	61	72	21	1A	07	00	CF	90	73	00	00	OD	00	00	00	Rar!Ï.s
mmma	0010h:	00	00	00	00	F7	CO	74	20	90	2C	00	OD	09	00	00	59	
	0020h:	22	00	00	02	3E	63	70	19	DA	59	В3	4A	ID	33	07	00	">cpY'J.3



看一下图片属性信息







Submit

1.4	11	与力	王	进	制车	专换		插件	-	妹子	F	其个	也功能	
填	写)	所常	5检:	则的	密起	Ъ:	(已	输入	字	符数	统	1:	179)	
• •	• •			•										
•		_											/	
▲ 结	果												1	
▲ 结 5B	果:	:	5490	B01	88F	52F	617	7070	920	0101	C		L	

直接解码获得flag 5BC925649CB0188F52E617D70929191C

Bugku-MISC-好多数值



我猜这应该是RGB数值

网上好多大佬解这题用的脚本是已经定义好X和Y,而我找的大佬脚本是自动进行因式分解,自动生成,可用性应该高一点

```
from PIL import Image
def Crack(n):#yinshufenjie
   flag = []
            print(each, int(n/each))
            flag += [(each,int(n/each))]
    if len(flag) == 1:return flag[0]
        return flag[int(choice)]
def Paint(X,Y,listrgb):#Draw according to string list
    pic = Image.new("RGB",(X, Y))
    i=0
    for x in range (0,X):
        for y in range (0,Y):
            temp = listrgb[i].split(',')
            pic.putpixel([x,y],(int(temp[0]),int(temp[1]),int(temp[2])))
   pic.show()
listrgb = open(r"/root/1.txt").readlines()
X,Y = Crack(len(listrgb))
Paint(X,Y,listrgb)
Paint(Y,X,listrgb)
```



flag{youc@n'tseeme}

Bugku-MISC-很普通的数独(ISCCCTF)

很普通的数独(ISCCCTF) 150

	zij	p						
	Flag						Sub	mit
附件有-	一个zip里面有25 1.zip - Ban	^{张图} dizip 7.04	(Professio	onal) 如图\\\\\\\	丁目(石)	https://blog ≢₽₽ħ/凵\		
]		, 」 」 解压	~	が いて 新建	上 兵 (1) + 添加	(11)		
	1.zip			名称 1.png 2.png 3.png 4.png 5.png 6.png 7.png 8.png		A https://blog	.csdn.net/we	ixin_47598409
252 	x 1 x 2 7 5 6 4 1 1 6 x 4 1 2 5 0 6 2 6 7 2 1 1 6 7 6 7 6 7 1 7 7 1 7 6 7 1 7 6 7 1 7 7 1		7 2 2 4 4 4 5 4 5 4 5 4 5 5 6 6 6 6 6 7 8 6 6 6 6 7 8 6 6 7 8 6 6 6 7 8 6 6 7 8 8 7 8 8 7 8 8 8 7 8 8 8 7 8 8 8 8 8 8 8 8 8 8 8 8 8		2 7 6 6 6 7 6 6 7 6 7 6 7 7 7 7 2 7 7 2 7 7 2 7 7 2 7 7 2 7 7 2 2 7 1 7 2 7 1 7 2 7 1 7 2 2 7 1 2 6 5 7 4 2 6 0 5 1			



全是数独图片

然后仔细看了一下1.png 5.png 21.png,好像是二维码的定位符,但是顺序乱了



然后把三张图片顺序调一下,5调到21,21调到1,1调到5

调完后,用记事本将信息记录下来,有数字的用1代替,没有的用0代替。

然后用py脚本把以上01转换为二维码



扫描二维码得出了一串base64编码



Vm0xd1NtUX1Wa1pPVldoVFlUSlNjRlJVVGtOamJGWnlWM

https://blog.csdn.net/weixin_47598409

Vm0xd1NtUXlWa1pPVldoVFlUSlNjRlJVVGtOamJGWnlWMjFHVlUxV1ZqTldNakZIWVcxS1IxTnNhRmhoTVZweVdWUkdXbVZHWkhOWGJGcHBWa1pa ZWxaclpEUmhNVXBYVW14V2FHVnFRVGs9

经过多次base64解码得出flag

文字加密解密 MD5加密/解密 URL加密 JS加/解密	JS混淆加密压缩	ESCAPE加/解密	BASE64	散列/哈希	迅雷,快车	E, 旋风URL加解密
1ag(y0ud1any1s1)		ZmxhZ3t5MHVkW	WFueTFzMX0=			
				口多行。日本	64神歌合人服	198964解密A 7 5 福空

flag{yOud1any1s1}

Bugku-MISC-PEN_AND_APPLE

PEN_AND_APPLE 150

狗师傅平日里比较害羞,但是又想追女神,于是他隐藏了一段信息在这段自拍中,这句话是他最想对女神说的话:)你能找到信息,并帮助狗师 傅表白成功么:)视屏在这儿:)

题目来源: 第七季极客大挑战

test.mp4

Flag

Submit

https://blog.csdn.net/weixin 47598409

这题原题的附件是一个rar压缩包里面有一个视频的,但是这个网站给的附件是一个test.mp4文件,应该被人修改了,所以就没做出来,不过可以在网上看到大概的解题思路

这题的flag为SYC{Hei_hei_hei}

Bugku-MISC-color

color 150

你见过彩虹吗?

来源: 第七届山东省大学生网络安全技能大赛



https://blog.csdn.net/weixin_47598409



有7张图片,看过每张图片的属性信息没可疑线索,用stegsolve打开











🛓 StegSolve 1	.3 by Caesu	ım	_	×
File Analyse Help				
Green plane 0				



🛓 StegSolve 1.3 by Caesum 🛛 –

 \times

File Analyse Help

Freen plane 1

根据7张图片给出的英文字母组成Make Me Tall——使我变高??? 还以为这个是一个flag,结果提交不了

那么 意思是让我变高 那么我们就在010或者winhex里面把它们都变高看看



然后发现每个图片最底层都是黑白相间的方块,看起来像是二进制。 使用记事本将黑白块用01记录下来。



仔细观察这几个二进制,发现第一列连起来1100110转换刚好对应的ASCII码是f

然后用大佬们的一个脚本跑一下,把每一列打竖的二进制进行转换



flag{Png1n7erEs7iof}

Bugku-MISC-怀疑人生



511

ctf2.jpg

ctf3.jpg

里面有三个文件,当我想解压ctf1.zip文件时要我输入密码。。。。。这。。

ZI

ctf1.zip

🔇 0% 已解压 ctf1.zip - Bandizip 7.04 (Professional)

0%		00:00:00	/ 00:00:0
	 輸入密码 	×]
0%	輸入加密文件的密码。		00:00:0
ctf.txt	ctf.txt		
	■ 輸入密码		Ŀ
脑		~	
对象			7598409

直接用我的专用破解压缩包加密神器PRTK,直接秒破,密码为password

A11					Properties Job Informatio	a
an e	Attack Type IIP dictionary attack	Status Finished	Result password (HEX	(=70617373776£7264]	Attack Type: Module: Profile: Status: Difficulty: Begin Time: End Time: Timeout After:	2IP dictionary attack 2IP Fassword Module English Finished Difficult 10/03/20 0:11:46 No Timeout
.c > 作务 名称	ž人生 > ctf1 ^	修改日期		类型		大小
📄 ct	f.txt	2018/12/	/7 17:44	文本文档		1 K
ctf.tx 文件(F)	t - 记事本 编辑(E) 格式(O) 查看(V) # Ilx1NmNcdTYxXHU2N1	骤助(H) 1x1N12IcdTV4X	HI I2MVx1	NiNcdT7		–

这个是base64编码

×

▲ 米斯特安全团队 CTFCrakTools pro v2.0 Beta
解码方式 进制转换 插件 妹子 其他功能
填写所需检测的密码: (已输入字符数统计: 60)
XHU2NIx1NmNcdTYxXHU2N1x1N2JcdTY4XHU2MVx1NjNcdTZiXHU2NVx1NzI=
结果:
\u66\u6c\u61\u67\u7b\u68\u61\u63\u6b\u65\u72
https://blog.csdn.net/weixin_47598409

得到 Unicode编码,直接Unicode转字符串

	│ 🎒 米斯特安全团队 CIFCraklools p	ro v2.0 Beta	
	解码方式 进制转换 插件	妹子 其他功能	
	填写所需检测的密码:(已输入:	字符数统计:44)	
2	\u66\u6c\u61\u67\u7b\u68\u	61\u63\u6b\u65\u7	2
	结米:		
	flag {hacker		

得出flag的第一部分

然后用010打开ctf2.jpg看一下

文件(F)	编辑	ŧ(E)	搜	索(S)	利	见图()	V)	格式	;(O)	脚	本(I)	樁	数	.) i	调试(D)	工具(T)	窗口(W)	帮助	I(H)
	-	-				۱ 🗖	÷.	1 👌	• 🖣		1	ົ	3	1 🌶	• 4	B 🍃	• >	A1 📐		
起始页	ci	f2 .	jpg	×																
▼ 编辑	访式	: +:	六进	制(H)	$) \sim$	运	行脚	本、	/ j	运行	模板	\sim	1							
	Ó			3		5	6		8			B	,Č	D	Ē		0123	456789A	BÇDEI	Ę
5A0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00				-
5B0h:	00	00	00	01	FF	D9	00	00	50	4B	03	04	14	00	01	00		ÿÙPK.		
5COh:	08	00	EC	96	87	4D	0B	2E	E9	DA	5A	00	00	00	37	01	ì-	‡MéÚZ	7	•
5D0h:	00	00	08	00	00	00	63	74	66	32	2E	74	78	74	б₽ìt	t ps:// b	log.cedn.	netwofxin		3409
	-	-	-					_	_			_		_	_			2-2		~

🤒 010 Editor - C:\Users\Administrator\Desktop\ccc\怀疑人生\ctf2.jpg

居然找到一个zip头,马上把这个文件的后缀名jpg改为zip

```
    ◇ ctf2.zip - Bandizip 7.04 (Professional)
    文件(F) 编辑(E) 查找(I) 选项(O) 视图(V) 工具(T) 帮助(H)
```

→ ⁶ ~ 打开 解压	下 新建	上 + 添加		۲ ۱
n ctf2.zip	名称 Ctf2.txt	1	^	
<pre> ctf2.txt - 记事本 文件(F) 编辑(E) 格式(O </pre>) 查看(V) 帮 ?.?!.? .?. .?!.? !!!!! !!! !.!!!.?	助(H) 	 ! ?!!.?	

得出这串东西,然后查了一下资料发现是ook编码 直接拿去解码,传送门https://www.splitbrain.org/services/ook

> ?!!.?? .?!.?! .?... ! 21 2. . .?!!! !!!!! !!!!? .?!.? !!!!! !!!!! !!!!! .?... 1711 ?!!.? I !?.?! .?!!! !!!!! !!!!! .!!!. ?.... .!?!! 11111 ?. . ?. ?! . ?!. ? . . . Text to Ook! Text to short Ook! Ook! to Text Text to Brainfuck Brainfuck to Text

3oD54e



得出flag的第二部分

ctf3.jpg这个看上去是一个模糊的二维码,直接用 QR Research直接可以扫描出



得出flag的最后一部分

即flag为flag{hackermisc12580}

Bugku-MISC-红绿灯



是一个gif的红绿灯图片,用stegsolve打开发现有1168张图片合成的一张gif图片





发现有一些图片是没有灯的,即灰色那些,红绿灯只有三种颜色,所以就把灰色(即没颜色的)删了,花了我15分钟才慢慢的删 完,累死我了

删除完之后,在文件夹里面调节大小的时候,突然发现最左侧和最右侧的一列颜色都是一样的,推测最右侧的一列为空格(即黄灯),因为ASCII码128位,最左侧的为0,得出绿色为0,红色为1,留下中间的7列。


一共455张图片,65组7位二进制数,红灯为1,绿灯为0,将图片信息记录下来

然后用py脚本把二进制转换为字符串



flag{Pl34s3_p4y_4tt3nt10n_t0_tr4ff1c_s4f3ty_wh3n_y0u_4r3_0uts1d3}

Bugku-MISC-不简单的压缩包



下载附件zip,修改一下文件名称



居然有密码。。。。

🚯 0% 已解压 11.zip - Bai	♦ 0% 已解压 11.zip - Bandizip 7.04 (Professional) - □ ×								
0%	輸入密码		×						
	輸入加密文件的密码。								
0%	flag.swf								
flag swf	输入密码								
11 zin - 该文件已揭坏			~						
	☑显示密码								
	密码管理器 确定	取消	7598409						

用010看了一下,发现这里有一个50 4B 03 04的十六进制zip文件头,而且还看到了50 4B的文件尾,应该是一个隐藏的zip文件 吧

O10 Editor - C:\Users\Administrator\Desktop\11.zip

文件(F)	编	髯(E)) #	叟索(S)	视图	(V)	格	بت ڑ(O) 8	却本(I)	模板	(L)	调词	式(D)	I	_具(T) i	窗口(W)	帮助)(H)			
	-	~			1		ø	1	80	h	R	5	9	1	۶	AB	5	+		A1			R	¶		0100 0001
起始页	1	1. z	ip>	<																						
₹ 编辑	方式	t: H	۲	性制(н) 🗸	1	运行	脚套	~_	运行	う模材	fç: z	IP.b	t ¥	Þ	_										
													B		Ď		F		12:	3456	78	PABC	DEF			
:3B40h	12	7D	C8	55	8B	9D	79	99	20	47	7D	7B	29	C6	19	5F	50		ÊU.	(.y ⁿ	• G	}{)Æ	. P			
:3B50h		4B	07	08	62	4A	96	8F	29	3B	06	00	DC	3A	06	00	50		κ	bJ		Ü:	P			
:3B60h	12	4B	01	02	1F	00	14	00	09	00	08	00	82	96	A3	4E	62		ζ				£Nb			
:3B70h	::	4A	96	8F	29	3B	06	00	DC	ЗA	06		08	00	24	00	00				Ü:		s			
:3B80h	.:	00	00	00	00	00	20	00	00	00	00	00	00	00	66	6C	61	П					fla			
:3B90h	:	67	2E	73		66	0A	00	20	00	00	00	00	00	01		18		J.SI	wf						
:3BAOh	::	00	F9	24	ЗB	ЗF	9E	01	D5	01	ЗB	DF	0A	E8	9E	01	D5		ù\$;?ž.	õ.,	:ß.è	ž.Õ			
:3BBOh	.:	01	3B	DF	0A	E8	9E	01	D5	01	50	4B	05	06	00		00		; ß	.èž.	õ.1	РК				
:3BCOh		00	01		01		5A	00		00	SF	ЗB		00			50	n I					P			
: 3BDOh	:: [4B	03	04	14	00	09	00	08	00	4A	95	A3	4E	65	1C	30		κ			J•£N	e.0			
:3BEOh		5E	2B	00	00	00	1A	00	00	00	0C	00	00	00	74	69	6E		`+.				tin			
:3BFOh	.:	67	73	68	75	6F	2E	74	78	74	AA	C5	51	DA	25	C6	68		sh	uo.t	xt	^a ÅQÚ	\$Æh			
:3C00h		42	FE	B6	90	58	63	2B	80	C5	12	7C	DC	Fl	бA	88	2F		Bþ¶	eXc+	ہ	. Üñ	j^/			
:3C10h	:	A1	A 5	60	32	B7	87	C7	DE	70	11	B1	66	FC	DE	EC	0F		¥1	2 · ‡Ç	p.	.±fü	Þì.			
:3C20h	:	11	80	FF	C3	50	4B	07	08	65	10	30	5E	2B	00	00	00		.€ÿź	ĂPK.		.0^+				
:3C30h	12	1A	00	00	00	50	48	01	02	1F	00			09			00	htte	b d/lb	PK.	Intelle	· 12 13 1 1	oction	1	7500	100
:3C40h		4A	95	A3	4E	65	1C	30	5E	2B	00	00	00	1A	00	00	00	nup	j•£1	Ne.C)^+		····	1_4	1098	409

在Kali用binwalk工具查看到这个压缩包里面有一个flag.swf文件和隐藏压缩包里面有一个tingshuo.txt文本

[rootaroot-09:51:37-Tony] → // ■ # foremost 11.zip Processing: 11.zip foundat=flag.swf2 �a⊡VS�Q&���:�dd 8\$♥♥♥\$S.♥♥♥♥/SQb1♥_V♥~♥♥ ~I♥♥7♥+?u♥j♥h♥♥♥ */ © foundat=tingshuo.txt♥Q♥%♥hB♥♥ */ rootaroot.09:53:30~Tony]	ŶŶĬŶŶŶjŶ7Ŷ8ĬŶŶŶŶ ĴŶ H?3ŶA6ŶASŶŶŶŶ)IB2ŶEŶ/ŶŶħŶŶŎŶŶ ~oH\$g-ŶŶŶŶŶŶŶŶŶ ŧŶŶ ŶŶjŶ/ŶŶ`2ŶŶŶ	Ih�s∭���₽���� ��U�#�;����010+ ��4�[�h�IMB®n& W��� �X ��l/<� �p�f����₽e0^+	ૡૼઙૺૹ૿ઌ૾ૡ૾ૡૹૼૹ૽ઌ૽ઌ૾ઌ૾ઌ ઌ ઌ૾૾ૺૺૺૺૺઌઌઌ૽ઌ૽૽ઌ૽ૺ ઌ૽ઌૺઌ૽૾ૺૺૺૺૺૺૺૺૡઌ૽ઌ૽ૺઌૺઌઌ૽ઌૺઌ૽ઌ૽ ઌઌ૽ઌ૽ઌૺઌઌ૽ૺઌ૾ઌ૱ઌ૽ૺૺૺૺૺૺ	H�6R Øa^w 4 ∲ØØ€
zip - 文件管理			_	
视图(V) 转到(G) 帮助(H)				
▲ /root/桌面/output/zip/			c	
超级帐户,可能会损害您的系统。				
00000000.zip 00000797.zip		https://bl	og.csdn.net/weixin_47	598409
$\leftrightarrow \rightarrow \uparrow$	位置(L): 🔤 /			
名称	▼ 大小	类型	修改日期	

÷	⇒	↑	位置(L): 🗖 /		
名称			-	大小	类型	修改日期
📄 tings	huo.txt			26 字节	plain text do	2019年5月3日18:42

然后用PRTK进行压缩包密码爆破,发现有文本的压缩包被秒破出来,密码是0,但是swf文件的那个压缩包还是没跑出来

408.3 kB

Shockwave ... 2019年5月3日 18:52

	😽 🧊 🔨				
iew All				Properties	
				Job Informatio	n
Job Name	Attack Type	Status	Result	Attack Type:	ZIP dictionary attack
000000	ZIP dictionary attack	Running		Module:	ZIP Password Module
0000797	ZIP dictionary attack	Finished	0 [HEX=30]	Profile:	English
				Status:	Finished
			T	Difficulty:	Difficult
				Begin Time:	10/03/20 9:57:30
			•	End Time:	10/03/20 9:57:31
				Timeout After:	No Timeout
				Decryptable:	Yes
				Result Type:	Password

打开文本,看到了一句日语

flag.swf



Bai 创翻译		官团	利力		翻译 API	同传一视频翻译
检测到日语 ▼	⇒	200+ 通种 中文(简体) ▼	翻	译	人工翻	译
パスワードは5	50桁だそ	うです			×	密码是50位

密码是50位? what?

然后根据全球知名解压缩工具bandizip给出的爆破密码的时间表



https://blog.csdn.net/weixin_47598409

密码50位,对应的应该是11.zip的,但是密码它有50位???暴力破解能解到地老天荒,密码会不会在属性里?然后看了一下, 什么线索都没有



常	规 安全	详细信息	以前的版本	
	属性	值		
	×14 名称	11.zip		
	类型	ZIP 压缩文件	‡	
	文件夹路径	C:\用户\Adr	ministrator∖≸	包面
	大小	399 KB		

然后来通过猜测一下密码,如果是50位混合组合密码,肯定想都别想了,要不试试用相同符号、数字、字母密码?

图 密码.txt - 记事本 文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

每一种都是相同的50个字符

打开ARCHPR,跑一下刚才制作的字典直接得出密码为50个a

ARCHPR 4.54 - 100%	- 🗆 X					
CH(F) W	升级 帮助 关于 退出					
加密的 ZIP/RAR/ACE/ARJ 文件 C:\Users\Administrator\Desktop\11.zip	攻击类型 空曲 口令已成功恢复!	×				
范围 长度 字典 明文 自动保存 字典选项	Advanced Archive Pas	sword Recovery 统计信息:				
字典文件路径:	11					
C:\Users\Administrator\Desktop\密码.txt	总计时间	27ms				
智能变化	平均速度(口令/秒)	407				
□尝试所有可能的大/小写组合	这个文件的口令	35666666666666666666666666666666666666				
■转换为 OEM 编码	十六进制口令	61616161616161616161616161				
状态窗口 2020/10/3 10:23:56 - 开始字典攻击		✔ 确定				
2020/10/3 10:23:56 - 口令已成功恢复! 2020/10/3 10:23:56 - 'ааааааааааааааааааааааа 是这个文件的一个有效口令	aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa					
当前口令: аааааааааааааааааааааааааааааааааааа	平均速度: 458 p/s 剩余时间:	STA				
1009	%					
ARCHPR version 4.54 (c) 1997-2012 ElcomSoft Co	o. Ltd.	https://blog.csdn.net/weixin_47598409				

这里敲一下黑板

基本信息

☞ 编辑

SWF是一种基于矢量的Flash动画文件格式,一般用FLASH软件创作并生成SWF文件格式,也可以通过相应软件将PDF等类型转换为SWF格式。SWF格式文件广泛用于创建吸引人的应用程序,它们包含丰富的视频、声音、图形和动画。可以在Flash中创建原始内容或者从其它Adobe应用程序(如Photoshop或Illustrator)导入它们,快速设计简单的动画,以及使用Adobe AcitonScript 3.0开发高级的交互式项目。设计人员和开发人员可使用它来创建演示文稿、应用程序和其它允许用户交互的内容。 Flash可以包含简单的动画、视频内容、复杂演示文稿和应用程序以及介于它们之间的任何内容。通常,使用Flash创作的各个内容单元称为应用程序,即使它们可能只是很简单的动画。您也可以通过添加图片、声音、视频和特殊效果,构建包含丰富媒体的Flash应用程序。

如何播放SWF:

可以利用FLASH控件实现播放FLASH的SWF文件,常用的第三方软件(如:实用Flash播放器、超级Flash播放器^[1]、 SWF Flash Player)可以直接在主流下载站下载后并安装,即可使用。

https://blog.csdn.net/weixin_47598409

因为我电脑没有安装什么flash播放器,所以运行不了flash动画的文件,但是我安装了一个可以查看swf源码的工具JPEXS Free Flash Decompiler,直接放进去进行代码审计



然后通过该工具的查找功能,查一下有没有flag或者Key等等关键词字

船 文本	搜索 ×
搜索文字:	К
	🗹 忽略大小写 📄 正则表达式
● 在AS中	□搜索 💿 在P-Code中搜索 💿 在文本中搜索
	确定 取消 1/159840



发现有这些文本里面有有K字符,然后逐个逐个的看 结果在第171个文本里面有一个K值,仔细看了一下是十六进制

	L甘	反混淆		
+	^ (AS源代码		
DefineButton2 (141)		1 stop():	~	-
DefineButton2 (191)		<pre>jun = new Array(_parent.COURSE_MAX);</pre>		1
⊕– [©] DefineSprite (24)		3 i = 0;		2
🟚–🏐 DefineSprite (43)		4 var k = "3F3F666C61677B6A7065787337726565666C6173687D2121";		2
DefineSprite (45)		5 while(i < _parent.COURSE_MAX)		3
DefineSprite (66)		6 {		4
DefineSprite (67)		7 c = 0;		5
DefineSprite (112)		a i = 0:		0
DefineSprite (173)		a while(i ≤ parent COURSE MAX)		1
DefineSprite (123)		10 {		8
E-C DefineSprite (140)	/	if(i = i)		9
- DefineSprite (142)				10
- DefineSprite (171)		if(must mont[i]) must mont[i])		11
🖆 📕 frame 1		13 II(_parent.record[1] / _parent.record[]])	1	12
DoAction	~	14 1		13
		15 c++;		14
基本标签信息		16 https://blog.csdn.net/weixin_4759	984	115
2称	伯	17		16

然后十六进制转ASCII码得出flag

16进制到ASCII字符串在线转换工具

1 3F3F6	66C61677B6A7	06578733772656	5666C6173687D		
₫ 清空	‡交换位置	示例转换	13保存结果	① 复制结果	
1 ??fla	g{jpexs7reef.	lash}!!			

flag{jpexs7reeflash}

Bugku-MISC-一枝独秀



https://blog.csdn.net/weixin_47598409



打开压缩包发现有一张png图片,于是就打开png图片一看,这。。。。

一枝独秀.png							
這一個一個一個一個	十 添加到	€	Ŵ	\heartsuit	ୢ	¥	※ 編

一枝独秀.png 似乎不支持此文件格式。

https://blog.csdn.net/weixin_47598409

用010或者winhex打开看一下

🤒 010 Editor - C:\Users\Administrator\Desktop\z\一枝独秀.png

3	文件(F)	编辑	計(E)	搜	索(S)	ł	见图(V)	格式	(O)	脚	本(I)	榰	total total	.) i	凋试((D)	工具(T) 窗口(W) 帮助(H)
, Ľ	\square \sim	-	-		1		۱ 🗖	÷	1 8	•		1	5	3	1)	• 1	B	🗩 🔶 I 🧛 📐 🎰 🕵 ¶ 🚦
1	起始页	-	枝独	誘.	png	x							475	-040 MG 8				
4	: 编辑	訪式	: +:	六进	制(H)		這	行脚	本、		运行	模板						
\$		4	+	?	3								B	ç	D	Ę		_0123456789ABCDEF
0	000h:	50	4B	03	04	14	00	00	00	00	00	4C	85	6C	4E	00	00	PKLIN
0	010h:	12	34	56	78	00	00	00	00	00	00	09	00	00	00	D2	BB	.4VxÒ»
0	020h:	D6	A6	B6	CO	DO	E3	2F	50	4B	03	04	14	00	01	00	08	Ö¦¶ÀÐã/PK
0	030h:	00	4C	83	6C	4E	A8	88	E7	22	6E	86	00	00	FD	8E	00	.LflN [~] ç"ntýŽ.
0	040h:	00	17	00	00	00	D2	BB	D6	A6	B6	CO	DO	E3	2F	66	6C	Ò»Ö¦¶ÀĐã/fl
0	050h:	6F	77	65	72	20	28	31	29	2E	6A	70	67	F2	DO	33	8A	ower (1).jpgòĐ3Š
0	060h:	04	B7	18	48	CE	24	05	33	A 7	B7	48	91	68	23	AE	74	
0	070h:	D6	3D	36	A8	59	47	28	AE	1A	45	1E	A3	15	BD	EC	Ettp	s://Bodread.cade.Bertweixi0.41298409

这不就是一个zip压缩包么?为何要改png格式。。。。。 于是把png改为zip,继续解压缩

� 0% 已解压 一枝独秀.z	ip - Bandizip 7.04 (Professional)	- 🗆 ×
0%	输入密码	×
100%	輸入加密文件的密码。	
	—枝独秀\flower (1).jpg	
ー枝独秀∖flower (1).jpg	輸入密码	
📕 🖪 🍢	hi	Iper//blog.code.net/weixin_47598400

啊。。。。。这。。。压缩包居然有密码?

打开压缩包仔细看了一下

Q ──枝独秀.zip ·	- Bandizi	p 7.04 (P	rofessiona	D)									- 0	×
文件(F) 编辑(E)	查找(I)	选项(0)	视图(V)	工具(T)	帮助(H)									
→ *	ł		P	+		B	\bigotimes	ij	¢	₽				82
打开	解压		新建	添加	劃除	測试	扫描	重要		時页				
🔒 一枝狼秀.zip		- 1	名称		^		压缩后大小		原始大小	类型	修改日期	加密算法	循环冗余检验((· ·
一枝狼秀			flower	(75).jpg*			34,414		36,605	JPG 文件	2019/3/12 16:26:23	ZipCrypto	22e788a	8
			flower	(76).jpg*			34,414		36,605	JPG 文件	2019/3/12 16:26:23	ZipCrypto	22e7884	8
			flower	(77).jpg*			34,414		36,605	JPG 文件	2019/3/12 16:26:23	ZipCrypto	22e788	8
		- 1	flower	(78).jpg*			34,414		36,605	JPG 文件	2019/3/12 16:26:23	ZipCrypto	22e788a	8
		- 1	flower	(79).jpg*			34,414		36,605	JPG 文件	2019/3/12 16:26:23	ZipCrypto	22e788a	8
		- 1	flower	(80).jpg*			34,414		36,605	JPG 文件	2019/3/12 16:26:23	ZipCrypto	22e788a	18
			a flower	(81).jpg*			36,411		43,725	JPG 文件	2019/3/12 18:39:57	ZipCrypto	014765	et
			G	1000			34.414		36 605	IDC WH	2010/2/12 16:26:22	Tefente	22.700.	0

发现第81张图片的CRC32跟其他的不一样,有猫腻

然后直接用上我最喜欢破解压缩包密码的神器PRTK

	🖲 🖹 🗊 🐐 🐥 🕷		
View All			
Job Name	Attack Type	Status	Result
枝独秀	ZIP dictionary attack	Finished	12345678 [HEX=313233343536373
State of the local state of the	and the second second		

密码秒破为12345678

解压玩之后打开第81张图片,看上去很普通,再看一下图片的属性



Flowers 这应该是一些线索吧

然后用stegsolve看一下



啊。。。。这。。有点像二维码,但是中间一大朵花挡住了,再翻几下也没有突破新的线索

然后前段时间打强网杯遇到了一题杂项题里面也是有一张jpg图片隐写了,当时是使用了JPHS这个工具进行查看jpg的隐写文件

打开软件后导入图片

JPHS for WIndows - Freeware version BETA test rev 0.5 - X

Exit Open jpeg Hide Seek Save jpeg Save jpeg as Pass phrase Options Help About

Vilesia 4	з КЪ	Width 960 g	pixel	Height 63	35 pix	el	
Approximat	e max	7 КЪ	reco	mmended		4 Kb	
			Hidden				
)irector Filename							
Filesi:	КЪ						
		S	aved jp	eg			
)irector Filename							
Filesi:	КЪ						

https://blog.csdn.net/weixin_47598409

然后点击Seek

it Open jpeg Hid	e Seek Savejpeg	Save jpeg as	Pass phrase	Options
elp About	4	<i>y</i> , 0		
	Input jpeg			
Director C: \Users\A	Administrator\Desktop	\z\一枝独秀\一枝	友独秀	
Filename flower (8)	.).jpg			
Filesis 43 Kb 1	Width 960 pixel Hei	aht 635 nivel		_
A Enter the pass phras	e and confirmation			×
****				_
D				
F protocococo				
F				
F				
F	ок	Cancel		
F	ОК	Cancel		

输入刚刚在图片属性里面的flowers



https://nie

取消

然后用010或者winhex打开隐写文件

保存类型(T):

all files (*.*)

🥦 010 Editor - C:\Users\Administrator\Desktop\z\xxx		nin	we trails.2	-		- 2.449000.	
文件(F) 編輯(E) 雅景(S) 祝園(V) 格式(O) 靜本(I) 模板(L) 満式(D) 工具(T) 窗口(W) 帮助(H)	🔺 🖷		1 私船快速方式	4 9	XI	们经经济同。	
I 🗈 🗸 📂 🖉 📾 📾 📾 🕼 I 🌭 📭 🏦 🗐 (Č) I 🔎 🚵 😓 -> I AI 📐 🚾 🕮 🖤	固定到"快 规制	和同志		移动到 复制到	動除 重命名	新建	居住 2 54
影始页 *** *	121/5119	光 剪切				又件类	@ m
▼ 编辑方式: 十六进制(H) × 送行標本 × 送行模板 ×		剪贴板		1	11月	新建	打开
0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF			20				
0000h: 50 4B 03 04 14 00 00 00 08 00 0A 83 6C 4E D8 34 EK	T	> 2	· · · · · · · · · · · · · · · · · · ·				
0010h: 09 C8 E0 01 00 00 B9 02 00 00 0C 00 00 00 B2 CE*							
0020h: CE F2 B7 F0 B7 A8 2E 74 78 74 2D 52 DB 72 A2 40 10.0		^					
0030h: 10 7D 4F 55 FE 66 3F 10 C9 8A 1A C5 4B 59 01 23 .)OUpf?.ÉS.ÅKY.#	★ 快速访问		and the second se				
0040h: 10 5D 16 50 C3 AA 58 94 06 05 F4 63 76 BA 67 78 .].FÅ*X"ocv*gx	-						
0050h: F2 17 72 A6 E2 4B D7 5C 4E 9F 4B CF 88 72 28 0A O.r: 4K*\NYKI'r(.		1					
0060h: 83 FD A3 7A 3D 88 62 2B FD 3E 75 33 9C 88 E2 53 fysz="b+y>u3g" 45	- 下戦	1				710	
0070h: 14 8E 7C 1B DC AB 41 63 FE 7D C0 76 47 51 7B EC .21.0#Acb}AvGQ(1						ZIP	
0080h; 74 80 54 2F F5 03 5C 1A E2 DC A5 49 78 AF 16 CF tET/6 AUVIX . I	1 又相	- *	_1016	No.	- 140	the size	
0090h; 4F A2 SE CA F4 9D FC F5 2F 8E 2B 39 FF CD CB 11 00*26,00/2+9v1E.	(国) 田片	*	0.0005		1002	304-op	
00A0h: 07 96 28 5E A5 7F 61 DF 53 37 9F 9D 33 38 29 31(^¥.aB57Y.38)1	Administ	trate #					
00B0h: F9 C3 A7 91 C9 E6 42 14 36 75 2E 1C C4 94 1D B8 uA\$ 'EeB.6uA".							

又发现了50 4B 03 04 的十六进制,这就是zip的文件头



立马修改后缀名

解压后发现一个txt文本





这。。。。。。佛语的意思不懂,再网上找到了这样的一个网站能进行佛语解码 网址http://www.keyfc.net/bbs/tools/tudoucode.aspx

把佛曰那句话复制进去

RSbbbVRZNm32W2X33mGm3Txt999RdV9hx0	
参悟佛所言的真意 🗲 2	普度众生
心不变,万物皆不变	
F那阿吳恋者曜立注醉波訂。訂厚住故訂罗醉恋咄烏語閣世 『冥依皤者哆諦故死哆夷菩侄曰呐逝至皤佛諳耶	1
the main the party that	参悟佛所言的真意 2 亦不变,万物皆不变 心不变,万物皆不变 諸迦亦侄栗侄大梵尼朋梵彌哆耨除怛奢般是諳爍悉哆燦晃 呐耨盧他姪明漫究呐得哆藐集能冥盡滅知俱朋怯室神奢顯 那阿冥悉奢薩豆涅缽波罰。罰摩侄故罰夢缽恐皤寫諳閣者 冥依皤者哆諦故死哆夷菩侄曰呐逝至皤佛諳耶

https://blog.csdn.net/weixin_47598409

点击参悟佛所言的真意,然后有一串东西 H-hDs10OZL3lhlZZbeRSbbbVRZNm32W2X33mGm3Txt999RdV9hx0

然后回想一下,题目给的提示

提示:翻过四个栅栏即可得到flag

栅栏密码加密解密

H-hDs100ZL31hIZZbeRSbbbVRZNm32W2X33mGm3Txt999RdV9hx0	
每组字数 4 加密 解密	
HINT-ZmxhZ3tDb29seW91R290SXROb3dZb3VLbm93VGhlRmxhZ30 https://blog.csdn.net/weixin_4	17598409

但是解完后又来一串东西。。。。感觉有点像是base64编码

然后用notepad++里面又一个插件可以解码

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□(W) ?	
🕞 🚽 🖶 🕞 🕞 🖕 🖌 🕦 🛅 🗩 c i 📾 🖕 🔍 🔍 🖾 🚍 🚍 1 🏋	MIME Tools >	Base64 Encode
new 1	Converter >	Base64 Encode with padding
1 HINT-ZmxhZ3tDb29seW91R290SXROb3dZb3VLbm93VGh1RmxhZ30	NppExport >	Base64 Encode with Unix EOL
A	插件管理	Base64 Encode by line
	打开插件文件本	Base64 Decode
· · · ·		Base64 Decode strict
		Base64 Decode by line
		Quoted-printable Encode
		Quoted-printable Decode
		URI Encode
☑ *new I - Notepad++ [Administration] 文件(F) 编辑(E) 搜索(S) 视图(V) 结果	tor] 编码(N) 语言	(1) 沿客(T) 丁目(0)
	b 9 C	
🕞 😑 🔚 🖻 🕞 🕞 🕞 📥 🛛 🔏 🐚 🛛	b 7 C	
□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	tNowYouKn	owTheFlag}

或者也可以来这里进行解码,传送门https://base64.us/

Base64.us Base64 在线编码解码 (最好用的 Base64 在线工具)
Base64 URLEncode MD5 TimeStamp
请输入要进行 Base64 编码或解码的字符
HINT-ZmxhZ3tDb29seW91R290SXROb3dZb3VLbm93VGhlRmxhZ30
编码 (Encode) 解码 (Decode) ‡ 交换 (编码快捷键: Ctrl + Enter)
Base64 编码或解码的结果: Sflag{CoolyouGotItNowYouKnowTheFlag} https://blog.csdn.net/weixin_47598409

得出flag{CoolyouGotItNowYouKnowTheFlag}

Bugku-MISC-小猪佩奇



作者: valecalida

zip	
Flag	Sub

https://blog.csdn.net/weixin_47598409

这题有点捞,网上的人都没人写WP,害得我解了几个小时找方法,难受一批

下载完后有一张png图



属性就大概这样子吧,没啥好看的,只发现是32位深度

	小猪佩哥	ភ <mark>ិ.</mark> png	属性	
۶	親 ਤ	全	详细信息	以前的版本
	属性 来源 - 拍摄日期 图像 -	朝	1	
	分辨率	6	30 x 359	
	宽度 高度	6 3	30 像素 59 像素	
	位深度	3	2	

首先来讲png图□,png图□是□种□损压缩的位图□形格式,也只有在□损压缩或者□压缩的图□(BMP)上实现lsb隐写。如果图 像是jpg图□的话,就没法使□lsb隐写了。原因是jpg图□对像数进□了有损压缩,我们修改的信息就可能会在压缩的过程中被破 坏。

用stegsolve打开,发现Green plane 0 和Blue plane 0,两张图那个小猪里面好像有一个二维码的线条,那就证明了这个题目是 lsb加密图片隐写



Bugku杂项小猪佩奇思路



7位纯字母

版权

去下载一个根据出题人说的字典

传送门:

https://github.com/danielmiessler/SecLists/blob/master/Passwords/darkweb2017-top1000.txt

但是python脚本的话我看到另外一个大佬写的WP

wind <

bugku-小猪佩奇 wp

ctf Pythonic misc 发布于 3月15日

一道带有LSB加密的图片隐写题,难度在于要跑密码,搜wp时看到出题人的提示:

https://blog.csdn.net/valecal...

字典: darkweb2017-top10000

密码:7位纯字母

写个python脚本跑出来就好了,我跑单线程时也挺快的(有提示可以少跑好多密码)

github上的脚本: https://github.com/livz/cloac...

需要的库自己pip一下就行

运行时用python2

代码:

#coding:utf-8 import threading from PIL import Image from lsb import extract

```
import re
with open('darkweb2017-top10000.txt','r+') as f:
    f=f.readlines()
filename='flag.png'
```

需要的库自己pip一下就行

运行时用python2

代码:

```
#coding:utf-8
import threading
from PIL import Image
from lsb import extract
import re
with open('darkweb2017-top10000.txt','r+') as f:
    f=f.readlines()
filename='flag.png'

def to_decode():
    for i in f:
        i=i.replace('\n','')
        if len(i)==7 and re.search('[0-9!?]',i)==None:
            out_file=i+'.txt'
            extract(filename,out_file,i)
to_decode()
```

将解出的N多个文件在linux下用file命令识别一下可以得到一个PNG,改下后缀名打开就是二维码,扫描 得flag https://blog.csdn.net/weixin_47598409

这位大佬提供的python用出题人说的字典进行爆破lsb隐写脚本,但是可能我太菜了,悟性不够高,运行了好几次这个脚本都是 出现这种情况



一直说我没有lsb库,我就用pip来安装,结果pip和pip3都是试过,依然不行,就这个问题花了我几个小时查资料还是没能解决, 我就没使用这位大佬的脚本了。



然后我就用github上提供的lsb脚本

https://github.com/livz/cloacked-pixel

根据脚本的使用教程,和出题人的提示说密码是纯7为字母,并且字典建议倒着用效果会更好,然后经过一番的排除操作



结果发现了正确密码就是raiders,中文意思是掠夺者,那应该代表就是获得flag的行为吧



然后就爆破出一张二维码



通过扫描二维码得出flag

OR Research	-	- 🗆	×
文件(F) 工具(T) 帮助(H)			
	纠错等级 H(30%) ▼ 版本 Auto Auto	推码 Auto 尺寸-	•
			^
			~
已解码数据 1:			^
位置:(60.3,32.3)-(323.0,32.3)-(60.3,295.0)-(323.0,294.9) 颜色正常,正像 版本:3 纠错等级:L,掩码:3 内容: flag{37d9704c-9752-434c-8891-ee15e1800490}			

flag{37d9704c-9752-434c-8891-ee15e1800490}

Bugku-MISC-好多压缩包



https://blog.csdn.net/weixin_47598409

打开压缩包里面有68个加密的压缩包

③ 123.zip - Ba 文件(F) 编辑(E)	ndizip 7.0 查找(I)	4 (Profe 选项(O	essional)) 视图(V)	工具(T) 帮	助(H)					_		×
→ ·	ß	~	F	+		4	\bigotimes					82
£177 € 122 zin	解症		新建	166JUL		测试	行油	自有	化码贝		140	
125.2ip			省称				压缩后大小	原始大	小英型		751	XD# "
			out0.zip	D			132	1	32 ZIP	达缩又件	20	16/8,
			out 1.zip	2			132	1	32 ZIP /	上海文1年	20	16/8,
			De out2.zip	2			132	1.	52 ZIP /		20	16/8,
			Dout3.zip	2			132	1.			20	16/8,
			Out4.zi	2			132	1	32 ZIP /		20	16/8
			Couto zip				132	1			20	16/0
			Couto.zip				132	1	2 710 5		20	16/0
			Out/.zip				132	1			20	16/0
			Couto.zip				132	1			20	16/0
			Cout10 a				132	1	2 710 5		20	16/0
			Cout11 a	ip in			132	1	2 710 5		20	16/9
			Cout12	ip in			132	1	22 710 5		20	16/9
			Cout13 7	in			132	1	2 7ID F	国文中	20	16/8
			aut14 z	in			132	1	22 7ID F		20	16/8
			aut15 7	in			132	1	22 7ID F	時代	20	16/8
			a out16 z	in			132	1	2 7IP F	国文件	20	16/8
			A out17 7	in			132	1	22 7ID F	日本	20	16/8
			A out187	in			132	1	22 7IP F	記念文仕	20	16/8
			A out10.2	in			132		22 7IP F	日宿文仕	20	16/8
			2				152				20	2
			< out19.z	эр			132	文件: 68,	52 ZIP 5	EMI文件 圧缩文件	20	> KB

然后随便点开一个压缩包看了一下发现被加密的压缩包里面有一个大小为4字节的文本



注意 一般情况,<mark>碰撞的字节数不会超过5(通常是3或者4字节)</mark>,否则要碰撞很久,碰撞时间太久的话 这个题就没什么意思了。一般看见压缩包里有很多文件,每个文件大小都小于5字节,才会用 crc32碰撞。

然后可以用CRC32碰撞脚本进行爆破

由于前段时间打强网杯的时候遇到了一题MISC题目,里面也需要CRC32碰撞,但是那个脚本只能单独的一个一个压缩包进行碰撞,虽然速度是秒破,但是一个一个的搞太麻烦了



然后在网上找到了大佬们的WP,他们写的CRC32碰撞脚本是直接把68个压缩包连在一起进行碰撞,然后把碰撞出来的内容保存在out.txt文本里面

这里有一点要注意,如果用的是python2来运行的话就必须要加上0xfffffff,如果是用python3就不用加要特别注意

if (binascii.crc32(str(i)) & 0xffffffff) == crc:

在 Python 2.x 的版本中, binascii.crc32 所计算出來的 CRC 值域为[-2^31, 2^31-1] 之间的有符号整数,为了要与一般CRC 结果作比对,需要将其转为无符号整数,所以加上& 0xfffffff来进行转换。如果是 Python 3.x 的版本,其计算结果为 [0, 2^32-1] 间的无符号整数,因此不需额外加上& 0xffffffff。

代码如下:





因为68个数量有点多,所以碰撞的话有点久

打开out.txt发现有一串base64编码

articus prana	/root/桌面/123/out.txt - Mousepad													
文件(F)	编辑(E)	搜索(S)	视图(V)	文档(D)	将助(H)									
	警告:您正在使用 root 账户,操作不当可能会损害您的系统。													
z5BzAAANAAAAAAAAAAKo+egCAIwBJAAAAVAAAAAKGNKv+a2MdSR0zAwABAAAAQ01UCRUUy91BT5UkSNPoj5hFEVFBRvefH														

于是用notepad++打开,并且用里面的插件进行解码

: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)	香件(P) 窗口(W) ?	
) 🗁 🖼 💫 🕼 🕼 🕼 🕼 🌔 🗩 🖒 📾 🧤 🔍 🔍 🖾 🖾 1 🗷	MIME Tools	> Base64 Encode
新建文本文档.txt⊠	Converter	> Base64 Encode with padding
1 z5BzAAANAAAAAAAAAAAAKo+egCAIwBJAAAAVAAAAAKGNKv+a2MdSR0zAwABAAAJ	NppExport	> Base64 Encode with Unix EOL
1	插件管理	Base64 Encode by line
	***	Base64 Decode
	打开瘤件又件类	Base64 Decode strict
81+(P) 图LI(W) ?		
3 📓 🖉 💿 🔍 🔍 🔳 🕨 🔛		

Flag.txt fix the file and get the flag -----里面有一个flag的文本,提示说要修复文件并且获得flag

然后我就用010打开这个base64解码后的文本

1	🤒 010 Editor - C:\Users\Administrator\Desktop\新建文本文档.txt																	
Ż	(件(F)	编辑	≩(E)	搜	索(S)	刮	1图(V)	格式	;(O)	脚	本(I)	梼	total total	.) i	周试	(D)	工具(T) 窗口(W) 帮助(H)
L.	` ~ [-			1			÷	1 🖠				5	3	13	•	B 🍃	🗅 🔶 I 🧛 📐 🔤
起	始页	顮	建文	(本)	て档.	txt	×											
Ŧ	编辑	访式	: +;	六进	制(H) ~	运	行脚	本、	1	运行	模板	~				~~~	
		Ó					5	é		8		Ă	B	Ç	Ď	Ē		0123456789ÅBCDEF
00	00h:	CF	90	73	00	00	0D	00	00	00	00	00	00	00	AA	ЗE	7A	Ï.sª>z
00	10h:	00	80	23	00	49	00	00	00	54	00	00	00	02	86	34	AB	.€#.IT†4«
00	20h:	FE	6B	63	1D	49	1D	33	03	00	01	00	00	00	43	4D	54	þkc.I.3CMT
00	30h:	09	15	14	CB	DD	41	4F	95	24	48	D3	E8	8F	98	45	11	ËÝAO•\$HÓè.~E.
00	40h:	51	41	46	F7	9F	1D	20	42	7C	6D	2B	B8	69	CA	9F	28	QAF÷Ÿ. B m+,iÊŸ(
00	50h:	2C	33	28	FC	48	16	99	lF	1B	18	1D	8F	38	2C	46	76	,3(üH.™8,Fv
00	60h:	El	C5	ED	67	4D	72	DE	4D	4A	D5	82	74	BE	92	BD	1F	áÅígMrÞMJÕ,t¾'⅓.
00	70h:	0A	94	CD	BE	AE	F7	ЗF	22	80	4A	F 7	74	20	90	2D	00	.″;®÷?"€J÷t .–.
00	80h:	1D	00	00	00	1D	00	00	00	02	62	Dl	E7	D5	4F	63	1D	bÑçÕOc.
00	90h:	49	1D	30	08	00	20	00	00	00	66	6C	61	67	2E	74	78	I.Oflag.tx
00	A0h:	74	00	BO	34	69	66	66	69	78	20	74	68	65	20	66	69	t.°4iffix the fi
00	B0h:	6C	65	20	61	6E	64	20	67	65	74	20	74	68	65	20	66	le and get the f
00	C0h:	6C	61	67	C4	ЗD	7B	00	40	07	00	I				ht	tps://bl	og.csdn.net/weixin_47598409

发现文件末尾是C43D7B00400700,不知道是什么东西,百度查了一下大佬们的WP说是rar文件尾,于是自己觉得不是很相信,就随便用一个rar文件导入进去看一下

🤒 010	Edito	or - (C:\Us	sers\	Adn	ninis	trate	or\D	eskte	op.							J.rar
文件(F)	编辑	≩(E)	搜	索(S)	刮	UB()	v)	格式	;(O)	脚	本(I)	樁	韨(L	.) i	调试	(D)	工具(T) 窗口(W) 帮助(H)
I 🗋 🗸	-			1			÷.	1 8	۰ I	۱ (5	3	1)	•	B	Þ 🔶 I 🧛 📐 🔤 🕵
起始页	Ť	新建3	<u>ረ</u> 本ን	て档.	txt			-			1	í	-	-		1	.rar X
▼ 编辑	訪式	: +:	六进(制()()) ~ (运	行脚	本、	/ 3	运行	模板	\sim					
	Ŏ			3		5	6		8			B	Ç	D	E		0123456789ABCDEF
D390h:	50	6F	EE	FF	7E	AB	44	CB	0E	20	5E	0C	4C	55	99	32	Poîÿ~≪DË. ^.LU™2
D3A0h:	2A	86	99	6F	E9	7B	EA	F2	7E	56	A9	BF	39	F1	EO	4A	*†™oé{êò~V©¿9ñàJ
D3B0h:	2C	FO	66	7D	76	69	6F	44	C4	ЗD	7B	00	40	07	00 htt	ps://bl	,ðf}vioDÄ={.@ olog.csdn.net/weixin_47598409

结果还真的是rar文件尾,然后我猜应该是要我们修复文件头吧



这个526172211A0700是rar的文件头

赾	始页	Ħ	建文	(本)	て档.	txt	×											
Ŧ	编辑	访式	: +:	六进	制(ਮ) ~	jā	行脚	本、	1	运行	模板	×		-	-		
ſ		Ŷ	÷	÷	~	7	~	Ŷ		. Y	2	÷	÷	4	ų	÷	÷	0123430703ADCDLT
oþ	00h:	CF	90	73	00	00	0D	00	00	00	00	00	00	00	AA	ЗE	7A	Ï.sª>z
ob	10h:	00	80	23	00	49	00	00	00	54	00	00	00	02	86	34	AB	.€#.IT+4«
00	20h:	FE	6B	63	1D	49	1D	33	03	00	01	00	00	00	43	4D	54	bkc.I.3CMT

只要我把rar头添加上去就可以了

起始页	顮	建文	(本)	て档	* ×												
▼ 编辑	方式	: +; 1	六进(_2	制(H) 3) \ 4)运 5	行脚 6	本、 7	้่่8	运行 [;] 9	模板 A	В	с	D	E	F	0123456789ABCDEF
0000h:	52	61	72	21	1A	07	00	CF	90	73	00	00	0D	00	00	00	Rar!Ï.s
0010h:	00	00	00	00	AA	ЗE	7A	00	80	23	00	49	00	00	00	54	ª>z.€#.IT

另存为rar文件



打开压缩包即可获得flag flag{nev3r_enc0de_t00_sm4ll_fil3_w1th_zip}

Bugku-MISC-一个普通的压缩包



https://blog.csdn.net/weixin_47598409

下载附件zip.rar

然后解压里面的flag.rar压缩包出现了这种情况
名称	^	修改日期	月	类型	大小				
flag		2020/1	0/2 17:27	文件夹					
flag.rar		2017/1	0/15 16:45	WinRAR archive		6 KB			
4	👠 WinRAR: Diagnostic m	essages							>
	Message		Archive						
	File header is corrupt: sec	ret.png	flag.rar	(C:\Users\Administrator\De	esktop\zip\	一个普通	的压缩	친\flag.rar)

Secret.png ???我猜应该里面有一张图

但是这样是解压出来是没有图片的,然后把刚刚题目的附件放进010或者winhex看了一下发现是50 4B 03 04 明显是zip文件头

🤒 010	Edito	r - (:\Us	ers\	Adn	ninist	rato	r∖De	eskto	op\z	ip.ra	r								
文件(F)	编辑	(E)	搜	索(S)	친	Q图(V) 1	格式	(O)	脚	本(I)	栘	製板(L) i	调试(C)) [工具(T)	窗口(\	N) 帮	助(H)
I 🗋 🗸	- ~			1				8		1	1 4	ົ	3	۱ 🌶	• 🗚	B 🍃	• >	I 🧛	📐 Hei	
起始页	zi	p. r	ar X	۲ ا																
▼ 编辑	方式:	;)	六进的	制(H)	¥	_j	行即。	本_ ~	'ì	至行:	模板	~						T		
	Ó	1	2	3		5			8	9		B	ç	Ď	Ę		0123	145678	9ABCD	EF
0000h:	50	4B	03	04	DA	00	00	08	00	00	E7	AB	51	4B	00 (00	PK		.ç«QK	
0010h:	00	00	00	00	00	00	00	00	00	00	19	00	00	00	Ehttp	s;// b	log,csdn	.net/wei	xin_475	98409 •

直接修改这个文件的后缀名,我这里改成是aaa.zip

再次用winrar解压flag.rar,还是出现这种情况

🥼 flag.txt - 记事本			3 31	
🚧 flag.rar	2017/10/15 16:45	WinRAR archive	6 KB	
📙 flag	2020/10/2 17:33	文件夹		

文本就只有一句话 flag is not here

然后把这个flag.rar文件放进010查看

页	fl	ag.	rar	×													
编辑	方式	: +; 1	六进(2	制(H) 3) ~ 4	运 5	行脚	本 >	้. ส	运行;	摸板 A	К	c	D	E	न	0123456789ABCDEF
)h:	AC	8C	9D	37	5A	C1	9A	D4	41	FC	44	B9	39	A4	8E	DE	⊣Œ.7ZÁšÔAüD¹9¤ŽÞ
)h:	8B	40	34	7E	18	91	59	ЗA	07	0E	D5	B7	68	81	A 8	DC	<@4~.`Y:Õ∙h.¨Ü
)h:	A 5	C9	72	B4	94	9D	AB	BO	68	A 8	F9	ЗE	15	95	AA	15	¥Ér'".«°h¨ù>.•ª.
)h:	DB	F6	9B	5A	8D	DA	C8	97	2F	E4	25	D8	66	80	1A	47	Ûö>Z.ÚÈ—/ä%Øf€.G
)h:	7B	El	59	7B	42	DD	AC	35	43	52	3C	E6	12	E6	6E	DC	{áY{Bݬ5CR<æ.ænÜ
)h:	BB	58	FB	CE	07	87	91	7F	FC	80	C4	ЗD	7B	00	40	07	»XûÎ.‡`.ü€ <mark>Ä={.@.</mark>
)h:	00														h	ttps://b	log.csdn.net/weixin_47598409

文件头和尾都正常,然后查了一些资料终于发现是这题涉及到rar文件的格式编码 详情参考https://blog.csdn.net/vevenlcf/article/details/51538837

如果用WinRAR的修复功能,则只能修复rar文件,不能把这个图片复原,rar格式编码中:

	未命名 — 已编辑
RAR压缩编码后的文件通常有以下	"数据块组成:
HEAD_TYPE=0x72	marker block【译者注: 有些文献里也称之为MARK_HEAD】 标志块【译者注: 一个固定为0x52 61 72 21 1A 07 00的7字节序列】
HEAD_TYPE=0x73	archive header【译者注: 有些文献里也称之为MAIN_HEAD】 归档头部块
HEAD_TYPE=0x74 类	file header【译者注:有些文献里也称之为FILE_HEAD】 文件块【译者注:直译为文件头部,但是此处的类型应该指的是整个块的类型,而非块头部结构的 型,因此感觉称之为文件块更合适。】
第三块为文件头,该数据块包含日 FILEATTR 的 CRC 结构和文件	函缩前的原文件大小、文件名、文件格式等信息,详细分析如下:块校验HEAD_CRC(2 字节)从 HEAD_TYPE 到 名。HEAD_TYPE(1 字节)头类型: 0x74。
HEAD_TYPE=0x75	old style comment header 老风格的 注释块【译者注:直译为注释头部,基于和文件块一样的原因,感觉称之为注释块更合适】
HEAD_TYPE=0x76	old style authenticity information 老风格的 授权信息块/用户身份信息块
HEAD_TYPE=0x77	old style <u>subblock</u> 老风格的 子块
HEAD_TYPE=0x78	old style recovery record 老风格的 恢复记录块
HEAD_TYPE=0x79	old style authenticity information 老风格的 授权信息块/用户身份信息块
HEAD_TYPE=0x7a	<u>subblock</u> 子块
HEAD_TYPE=0x7b	end block 结束块【译者注: 一个固定为0xC4 3D 7B 00 40 07 00的7字节序列】//blog.csdn.net/weixin_47598409

然后根据这个编码的解释再来分析一下这个rar文件

颌	£	lag.	rar	×													
编辑	訪式	: +:	六进	制(ਮ)) ~	jź	行脚	本、	1	运行	模材	Y	-		100	<u>, 112 - 1</u>	
	Q		2	3		5	6	7	8	9	Ą	B	Ç	D	Ę		0123456789ABCDEF
Oh:	52	61	72	21	1A	07	00	CF	90	73	0.0	20	0D	00	00	00	Rar!Ï.s
Oh:	00	00	00	00	D5	56	74	20	90	2D	00	10	00	00	00	10	ÕVt
Oh:	00	00	00	02	C7	88	67	36	6D	BB	4E	4B	1D	30	08	00	Ç^g6m»NK.0
Oh:	20	00	00	00	66	6C	61	67	2E	74	78	74	00	BO	57	00	flag.txt.°W.
Oh:	43	66	6C	61	67	20	69	73	20	6E	6F	74	20	68	65	72	Cflag is not her
0h:	65	A 8	3C	7A	20	90	2F	00	3A	15	00	00	42	16	00	00	
Oh:	02	BC	E9	8C	2F	6E	84	4F	4B	1D	33	0A	00	20	00	00	Her n, OK 3

红色框圈住74就是rar里面的flag.txt文本头部编码,这个文本的十六进制一直从00到65 才结束,这样可以判断出74是在rar文件的格式编码的一种文件数据块头部,里面包含文 件大小、文件名、信息等等

由此可见rar里面还有一张名为Secret.png的图,在flag.txt结束的地方就是下一个块开始的地方,所以png文件的头部编码为74,我们修改过来即可:

ī	£	lag.	rar	×													
赭	方式	: +;	六进的	制(H) ~	运	行脚	本、	1	运行;	模板	~					
	Ó		2	3		5	6	7	8	9	A	B	Č	D	Ę		0123456789ABČDEF
13	52	61	72	21	1A	07	00	CF	90	73	00	00	0D	00	00	00	Rar!Ï.s
11	00	00	00	00	D5	56	74	20	90	2D	00	10	00	00	00	10	ÕVt
.:	00	00	00	02	C7	88	67	36	6D	BB	4E	4B	1D	30	80	00	Ç^g6m»NK.0
::	20	00	00	00	FS	6C	61	67	2E	74	78	74	00	B0	57	00	flag.txt.°W.
::	43	66	6C	61	57	20	69	73	20	6E	6F	74	20	68	65	72	Cflag is not her
13	65	A 8	3C	74	20	90	2F	00	ЗA	15	00	00	42	16	00	00	e [~] <t .="" .:b<="" td=""></t>
12	02	BC	E9	86	2F	6E	84	4F	4B	1D	33	0A	00	20	00	00	.4éŒ/n"OK.3
::	00	73	65	63	72	65	74	2E	70	6E	67	00	FO	40	AB	Htps:/	/blogfc56ff.fret/weikfn_47598409
	11	011	11	E E	00	D 1		00	on	00	04	00	07	00	22	10	A TT ATTAC THE AND

保存后就可以正常的用winrar解压缩



图片终于出来了

但是很遗憾这个图片是一张空白图片,很不甘心,直接把图片放进010或者winhex查看

起始页	56	ecre	t. pı	ng X		/											
▼ 编辑	方式	: +;	六进的	制(光)	M	运	行脚	本、	1	运行	模板	~					X
	0	1	2	3	+	5	Ģ		8	9		B		Ď	Ę		0123456789ABCDEF
0000h:	47	49	46	38	39	61	18	01	18	01	91	02	00	FE	FF	FF	GIF89a`þÿÿ
0010h:	FF	FF	FF	FF	FF	FF	00	00	00	21	FF	0B	58	4D	50	20	ÿÿÿÿÿÿ!ÿ.XMP

)疳包 → flag



https://blog.csdn.net/weixin_47598409

但是改了后缀还是空白图,然后我就用ps打开这个文件,发现有两个白色图层



然后进行每个图层单独分离为png文件,共两个

3	て件	编辑	图像	图层	文字	选择	滤镜	3D	视图	窗口	帮助			
	新建				ЖN									
	打开				жo			-	-					
	在B	ridae 🛱	词道的		T #0									
	打开	为智能	对象		2000		示变换控	件	- +	= =	T	₩ <u>h</u>	•••	
	最近	打开文	件		<u> </u>	B/8)*								
	关闭				жw	/								
	关闭	全部			~ #W	1								
	关闭	其它			Σ₩P									
	关闭	并转到	Bridge.		<mark>ት</mark> װ	1								
	存储				жs									
	存储	为			<mark>ሰ</mark> ፝፞፝ <mark>ដ</mark> Տ									

	540		
/阦复	F12		
导出		快速导出为 PNG	
生成	•	导出为	V ж 0 ፓ
ー ハ 孝… 在 Behance 上共享…		导出首选项	
) 搜索 Adobe Stock		存储为 Web 所用格式(旧版)…	ጚዕ <mark></mark> ജs
置入嵌入对象 置入链接的智能对象 打包		导出为 Aero 画板至文件 将画板导出到 PDF	
自动		将图层导出到文件	
脚本		将图层复合导出到 PDF https://blog	.csdn.net/weixin_47598409



https://blog.csdn.net/weixin_47598409



然后我用stegsolve分别打开这两个图

然后把两个图进行合拼



这个时候扫描得不出什么信息,二维码的定位符都不完整,左上角的就不说了,没法补上,右上角可以补上黑方块:





https://blog.csdn.net/weixin_47598409

然后使用二维码查看工具QR Research直接扫描就可以得到flag



flag{yanji4n_bu_we1shi}

Bugku-MISC-2B





一个这样的文件不知道是什么东西,用010打开看了一下是PNG文件

010 Editor - C:\Users\Administrator\Desktop\2B

て(牛(F)	编辑	(E)	搜	素(S)	刮	见图(\	Ŋ	格式	(O)	脚	本(I)	枝	靓(L	.) i	调试	(D)	I	l(T)	窗	₽D(V	V)	帮助(H)
\sim							÷	1 🖠				5	3	13	• ,	в I	P	÷	14	M .		Hex 💕
始页	28	X																				
编辑	;方式:	: +:	六进制	制(H)	1	运	行脚	本 ~	1	运行	模板	~						. /	1			
	Ŏ		2	3 🖡	4	5	6		8	9	A	B	Ċ	Ď	E		C	1	345	678	9AB	CDEF
000h:	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	. 1	PN(G.,		••••	IHDR
10h:	00	00	07	80	00	00	04	B 0	08	02	00	00	00	95	5 ^b 2tt	ps://b	log.c	sdn	eet/v	veixir	1_47	598409

改后缀,打开图片,这。。。。。。。看过图片属性没东西



用binwalk看到了一个zip文件隐藏了

(rooti)root	~ 10:48:21~Tony] ~/桌面# binwalk	2B.png
DECIMAL 800	HEXADECIMAL	DESCRIPTION
0 2066228 ressed size: 5131827	0×0 0×1F8734 3099670, name: 0×4E4E33	PNG image, 1920 x 1200, 8-bit/color RGB, non-interlaced Zip archive data, at least v2.0 to extract, compressed size: 3065475, uncomp B2.png End of Zip archive, footer length: 22

用foremost进行分离



但是在Kali解压出现了错误



在window系统里面进行解压,发现它要我输入密码

🔇 0% 已解压 00004035.zip	- Bandizip 7.04 (Professional)		×	
0%	 輸入密码 		×	<
0%	輸入加密文件的密码。			
	B2.png			_
B2.png	輸入密码			
🖪 🖪 🖓 🌄 =			~	
				409

现在只有一种想法,就是这个是一个伪加密ZIP,因为在win打开要输入密码,在Kali打开直接报错,如果正常有密码的话,在两种系统都能正常要你输入密码的,为了验证是不是伪加密,我使用010进行修改一下试试

				. .	_		· •	- T			1	`	1	- 1	۱	~	1	
起始页 🛛 🛛	0004	035.	zip	x														
▼ 编辑方式	t: +:	六进	制(H) ~	运	行脚	本、	1	运行	模板	: ZI	P.bt	~	Þ				-
1	Ó		2					7				B	Ç	Ď	Ę			0123456789ABCDEF
E:C690h:	79	99	A 7	EC	DD	41	64	DF	19	E5	lF	AC	8F	F9	7D	4B	4	y™§ìÝAdß.å.⊣.ù}K
E:C6A0h:	70	AD	Fl	9F	C9	FF	07	50	4B	01	02	lF	00	14	00	01		p-ñŸÉÿ.PK
E:C6B0h:	00	08	00	DO	00	BE	4C	A 4	20	76	Bl	83	C6	2E	00	16		Ð.%L¤ v±fÆ
E:C6C0h:	4C	2F	00	06	00	24	00	00	00	00	00	00	00	20	00	00		L/\$
E:C6D0h:	00	00	00	00	00	42	32	2E	70	6E		0A	00	20	00	00		B2.png
E:C6E0h:	00	00	00	01	00	18	00	FD	82	C6	02		F7	DЗ	01	7E		ý,Æ.g÷Ó.~
E:C6F0h:	BC	98	DF		F7	D3	01	08	C5	D7	5D		F7	DЗ	01	50		₩~ßg÷ÓÅ×]g÷Ó.P
E:C700h:	4B	05	06	00	00	00	00	01	00	01	00	58	00	00	00	A 7		K§
E:C710h:	C6	2E	00	00	00										ł	nttps	://b	log.csdn.net/weixin_47598409

结果发现蓝色选中的那两个十六进制的数为0100,这个就是伪加密,如果想要把伪加密移除的话要改为0000



直接解压成功

打开里面发现有一张为B2.png的图片,看上去和题目给的附件的图片好像是一样的,然后我去看一下两张图的属性信息对比一下

corbing value	L	^	■ B2.png ⊯1	r.	
规安全	详细信息 以前的版本		常规 安全	详细信息 以前的版本	
属性	值	^	属性	值	^
来源			来源		- 1
拍摄日期			拍摄日期		
图像			图像		
分辨率	1920 x 1200		分辨率	1920 x 1200	
宽度	1920 像素		宽度	1920 像素	
高度	1200 像素		高度	1200 像素	
位深度	24		位深度	24	
文件			文件 ——		
名称	2B.png		名称	B2.png	
项目类型	PNG 文件		项目类型	PNG 文件	
文件夹路径	C:\用户\Administrator\桌面		文件夹路径	C:\用户\Administrator\桌面\00004035	
创建日期	2020/10/3 10:41		创建日期	2018/5/30 0:09	
修改日期	2020/10/3 10:41		修改日期	2018/5/30 0:06	
大小	4.89 MB		大小	2.95 MB	
属性	A		属性	https://blog.csdn.net/weixin_	4759840

结果发现两站图就是除了名字和大小不一样之外,其他的属性都一样的,我猜应该是图片盲水印攻击,因为前段时间打强网杯的 时候也是遇到这种情况,所以有了经验

盲水印攻击脚本下载传送门https://github.com/linyacool/blind-watermark

然后把两张图都放在这个文件夹里面,用python2环境跑一下,注明一下B2是无水印图片,2B是有水印图片

[root@root~15	:12:	43~Tony] /blind-watermar	k# python deco	ode.pyorig	inal B2.png	image	2B.png	result	flag.png
NG 图像,1920×	×	+				N.			
' 	6	file:///root/桌	面/blind-wate	ermark/flag.p	ong				••• 🗟 •
k 🔨 Kali Trainir	ng	🔨 Kali Tools	🧧 Kali Docs	器 百度一下	,你就知道	🥆 Kali I	orums	Net	Hunter J
	U	ISTR		<u></u>	2B	THO ttps://blog.	nB csdn.ne	t/weixin_4	47598409

这里要吐槽这个flag,真的恶心,那个F后面的看上去真的很像一个0,但却是一个大写O,那个前括号后面的是i的大写不是L的小 写

NUST{I_10v3_2B_FOr3v3r}

Bugku-MISC-QAQ



QAQ 200

by LordCasser 原创 此题投稿原创题,writeup尽量不要发网上

暂时不放Tips



https://blog.csdn.net/weixin_47598409

下载两个附件

🗐 ciph	er.txt - ic	事本		
文件(F)	编辑(E)	格式(O)	查看(V)	帮助(H)
FSAnR	AIZNIN	IjPQMj	NyBJNT	s6NIIFPFIqDDVTJy0zGE8rKxZBJDIrJkYoPUQML1M3MDYJZTElFyI7
UzE6D	TtSNxc	kNDw2	Mxk9Jz	C=

1

文本是一串base64编码拿去解码一下

Base64.us Base64 在线编码解码	(最好用的 Base64 在线工具)
Base64 URLEncode MD5 TimeStamp	
请输入要进行 Base64 编码或解码的字符	
FSAnRAIzNIMjPQMjNyBJNTs6NIIFPFIqDDVTJy(<u>UzE6DTtSNxckNDw2Mxk9Jzc</u> =	0zGE8rKxZBJDIrJkYoPUQML1M3MDYJZTEIFyI7
编码 (Encode) 解码 (Decode) ↓交换	(编码快捷键: Ctrl + Enter)
Base64 编码或解码的结果:	
□ 'D□36S#=□#7 I5;:6R□ <r*5s'-3□o++□a\$2+8 ;R7□\$4<63□='7</r*5s'-3□o++□a\$2+8 	&F(=D/S706 e1%□";S1:

有点是乱码

然后把QAQ附件放进去010分析

포도외	<u>nw</u>	41	ra v															
¥	编辑	方式	: +;	六进的	制(H)) ~	运	行即	本 〜	· ;	运行,	摸板	~					
		Ŏ											B		Ď	Ę		0123456789ABCDEF
DOA	0h:	83	01	41	64	83	01	37	33	7D	02	71	12	57	34	7C	02	f.Adf.73}.q.W4 .
DOB	Oh:	53	64	29	02	4E	DA	00	29	04	DA	05	72	61	6E	67	65	Sd).NÚ.).Ú.range
DOC	Oh:	DA	03	6C	65	6E	DA	03	63	68	72	DA	03	6F	72	64	29	Ú.lenÚ.chrÚ.ord
DOD	Oh:	04	DA	03	6B	65	79	DA	05	70	6C	61	69	6E	DA	06	63	.Ú.keyÚ.plainU.c
DOE	Oh:	69	70	68	65	72	DA	01	69	A9	00	72	0A	00	00	00	FA	ipherÚ.i©.rú
OOF	0h:	06	51	41	51	2E	70	79	DA	06	65	6E	63	72	79	74	01	.QAQ.pyU.encryt.
010	Oh:	00	00	00	F3	08	00	00	00	00	01	04	01	12	01	2C	01	···ó····,·
011	Oh:	72	0C	00	00	00	63	00	00	00	00	00	00	00	00	03	00	rc
012	Oh:	00	00	09	00	00	00	43	00	00	00	73	ЗA	00	00	00	64	Cs:d
013	Oh:	01	7D	00	74	00	64	02	83	01	8F	24	7D	01	78	1C	7C	.}.t.d.f\$}.x.
014	0h:	01	6A	01	83	00	7D	02	7C	02	72	26	7C	00	7C	02	37	.j.f.}. .r& . .7
015	Oh:	69	7D	00	71	10	50	6E	71	10	57	34	57	6C	64	00	51	i}.q.Pnq.W4Wld.Q
016	Oh:	6C	52	79	58	5F	7C	00	53	7A	29	03	4E	72	01	00	00	lRyX_ .Sz).Nr
017	0h:	00	FA	09	70	6C	61	69	6E	2E	74	78	74	29	02	DA	04	.ú.plain.txt).Ú.
018	0h:	6F	70	65	6E	DA	08	72	65	61	64	6C	69	6E	65	29	03	openÚ.readline).
019	0h:	72	07	00	00	00	DA	01	66	DA	04	6C	69	6E	65	72	0A	rÚ.fÚ.liner.
DIA	0h:	00	00	00	72	0A	00	00	00	72	0B	00	00	00	DA	00	67	rÚ.g
01B	0h:	65	74	50	6C	61	69	6E	54	65	78	74	07	00	00	00	F3	etPlainTextó
01C	Oh:	10	00	00	00	00	01	04	01	0A	01	02	01	08	01	04	01	
01D	0h:	0A	02	10	01	72	13	00	00	00	63	00	00	00	00	00	00	rc
DIE	0h:	00	00	04	00	00	00	0A	00	00	00	43	00	00	00	73	3E	Cs>
DIF	Oh:	00	00	00	64	01	7D	00	74	00	83	00	7D	01	74	01	7C	d.}.t.f.}.t.
020	0h:	00	7C	01	83	02	7D	02	74	02	64	02	64	03	83	02	8F	. .f.}.t.d.d.f
021	0h:	16	7D	03	7C	03	6A	03	7C	02	6A	04	64	04	83	01	83	.}. .j. .j.d.f.f
022	Oh:	01	01	66	57	6C	64	00	51	61	52	67	58	7B	64	00	53	fWld.QaRgX{d.S
023	0h:	66	29	05	4E	DA	0A	4C	6F	72	64	43	61	73	73	65	72	f).NU.LordCasser
024	Oh:	FA	0A	63	69	70	68	65	72	2E	74	78	74	DA	01	77	DA	ú.cipher.txtŰ.wŰ
025	Oh:	07	62	61	73	65	5F	36	34	29	05	72	13	00	00	00	72	.base_64).rr
026	Oh:	0C	00	00	00	72	OF	00	00	00	DA	05	77	72	69	74	65	Ū.write
027	0h:	DA	06	65	6E	63	6F	64	65	29	04	72	06	00	00	00	72	Ú.encode).rr
028	Oh:	07	00	00	00	72	08	00	00	00	72	11	00	00	00	72	0A	rr.
029	Oh:	00	00	00	72	0A	00	00	00	72	OB	00	00	00	DA	04	6D	rU.m
02A	Oh:	61	69	6E	12	00	00	00	F3	0A	00	00	00	00	01	04	01	ainó
02B	0h:	06	01	0A	01	0C	01	72	18	00	00	00	DA	08	5F	5F	6D	TUm
02C	Oh:	61	69	6E	5F	5F	4E	29	04	72	0C	00	00	00	72	13	00	ain_N.rr.
02D	Oh:	00	00	72	18	00	00	00	DA	08	5F	5F	6E	61	6D	65	5F	rUname
02E	Oh:	5F	72	AO	00	00	00	72	AO	00	00	00	72	AO	00	00	00	_rr.
D2F	Oh:	72	0B	00	00	00	DA	08	3C	6D	6F	64	75	6C	65	3E	01	rU. <module>.</module>
查找	结果																https:/	/blog.csdn.net/weixin_4759840

看到chr、ord、range、QAQ.py、main、name基本可以判断是个pyc文件

添加后缀名为pyc文件后用uncompyle6进行pyc反编译

pip install uncompyle6 #安装



得到QAQ.pyc反编译后的tt.py脚本



然后根据以上的加密脚本进行编写一个解密脚本pp.py

```
cipher += chr(ord(key[i % len(key)]) ^ ord(plain[i]))
def getPlainText():
   break
return plain.decode('base_64')
plain = getPlainText()
clear = decryt(key,plain)
 f.write(clear)
if ___name___ == '___main___':
```

得到clear.txt文本是一些明文



YOU ARE FOOLED THIS IS NOT THAT YOU WANT GO ON DUDE CATCH THAT STEGOSAURUS	×	你被愚弄了 这不是你想要的 去吧,伙计 抓住剑龙
(1)) イヤ		〔小〕 [音] 报错 笔记▼
见一个单词STEGOSAURUS——剑龙? ?		

Python隐写工具

用于在Python字节码中嵌入有效负载的隐写工具。

Stegosaurus是一种<u>隐写术工具</u>,允许在Python字节码(pyc或pyo)文件中嵌入任意有效载荷。嵌入过程不会改变运营商文件的运行时行为或文件大小,并且通常会导致编码密度较低。有效负载分散在整个字节码中,因此工具 strings 不会显示实际的有效负载。 dis 在使用 Stegosaurus嵌入有效负载之前和之后,Python的模块将返回相同的字节码结果。此时,没有 关于这种类型的有效载荷传递的先前工作或检测方法。

剑龙需要Python 3.6或更高版本。

用法



https://blog.csdn.net/weixin_47598409

然后去下载stegosaurus脚本 传送门https://github.com/AngelKitty/stegosaurus

执行这个脚本,分离出flag



flag{fin4lly_z3r0_d34d}

科普一下这个stegosaurus的知识和用法

提示告诉我们用|SIEGOSAURUS|工具进行隐写的,我们直接将隐藏的payload分离出来即 可。 python3 stegosaurus.py -x QAQ.pyc (11/29/18@ 4:14下午)(python@Sakura):~/桌面 python3 steqosaurus.py -x OAO.pyc Extracted payload: flag{fin4lly z3r0 d34d} 我们得到了最终的 flag 为: flag{fin4lly z3r0 d34d} 既然都说到这个份子上了,我们就来分析—下我们是如何通过|Stegosaurus|来嵌入 Payload . 我们仍然以上面这个代码为例子,我们设置脚本名称为 encode.py 。 第一步,我们使用|Stegosaurus|来查看在不改变源文件|(Carrier)|大小的情况下, 我们的|Payload|能携带多少字节的数据: python3 -m stegosaurus encode.py -r (11/29/18@ 4:22下午)(python@Sakura):~/桌面 python3 -m stegosaurus encode.pv -r Carrier can support a payload of 24 bytes 现在,我们可以安全地嵌入最多24个字节的 | Payload | 了。如果不想覆盖源文件的话, 我们可以使用 --> 参数来单独生成一个嵌入了 Payload 的 py 文件: python3 -m stegosaurus encode.py -s --payload "flag{fin4lly z3 (11/29/180 4:24下午)(python@Sakura):~/桌面 python3 -m stegosaurus encode.py -s --payload "flag{fin4lly_z3r0_d34d}" Payload embedded in carrier 现在我们可以用 18 命令查看磁盘目录, 嵌入了 Payload 的文件(carrier 文件) 和原始的字节码文件两者大小是完全相同的: 现在我们可以用 1s 命令查看磁盘目录, 嵌入了 Payload 的文件(carrier 文件) 和原始的字节码文件两者大小是完全相同的: (11/29/180 4:34下午)(python@Sakura):-/桌面 ls -l __pycache__/encode.cpython-36.pyc rw-r--r-- 1 python python 785 11月 29 16:30 __pycache__/encode.cpython-36.pyc (11/29/180 4:35下午)(python@Sakura):-/桌面 ls -l __pycache__/encode.cpython-36-stegosaurus.pyc rw-r--r-- 1 python python 785 11月 29 16:30 __pycache__/encode.cpython-36-stego saurus.pyc

注:如果没有使用 -3 参数,那么原始的字节码文件将会被覆盖。

python3 -m stegosauruspycache/encode.cpython-36-s ((11/29/18@ 4:35下午)(python@Sakura):~/桌面 python3 -m stegosauruspycache/encode.cpython-36-stegosaurus.pyce Extracted payload: flag{fin4lly_z3r0_d34d} 我们构造的 Payload 不一定要是一个 ASCII 字符串, shellcode 也是 (11/29/18@ 4:39下午)(python@Sakura):~/桌面 python3 -m stegosaurus <u>encode.py</u> -spayload "\xeb\x2a\x5e\x89\x76" Payload embedded in carrier	
<pre></pre>	tegosaur
<pre>#(11/29/18@ 4:35下午)(python@Sakura):~/桌面 python3 -m stegosauruspycache/encode.cpython-36-stegosaurus.pyc Extracted payload: flag{fin4lly_z3r0_d34d} 我们构造的 Payload 不一定要是一个 ASCII 字符串, shellcode 也是 #(11/29/18@ 4:39下午)(python@Sakura):~/桌面 python3 -m stegosaurus encode.py -spayload "\xeb\x2a\x5e\x89\x76" Payload embedded in carrier</pre>	•
我们构造的 Payload 不一定要是一个 ASCII 字符串, shellcode 也是 (11/29/180 4:39下午)(python@Sakura):~/桌面 python3 -m stegosaurus <u>encode.py</u> -spayload "\xeb\x2a\x5e\x89\x76" Payload embedded in carrier	; -x
#(11/29/18@ 4:39下午)(python@Sakura):~/桌面 python3 -m stegosaurus <u>encode.py</u> -spayload "\xeb\x2a\x5e\x89\x76" Payload embedded in carrier	可以的:
<pre>#(11/29/18@ 4:42下午)(python@Sakura):~/桌面 python3 -m stegosauruspycache/encode.cpython-36-stegosaurus.pyc Extracted pavload: \xeb\x2a\x5e\x89\x76</pre>	- ×
https://blog.csdn.net	/weixin_47598/

```
import sys
 import os
 import math
 def add(a,b):
    return int(a)+int(b)
 def sum1(result):
     return int(result)*3
 def sum2(result):
     return int(result)/3
 def sum3(result):
     return int(result)-3
 def main():
    a = 1
    b = 2
    result = add(a,b)
    print(sum1(result))
     print(sum2(result))
     print(sum3(result))
 if __name__ == "__main__":
     main()
我们让它携带 Payload 为 flag_is_here 。
```

https://blog.csdn.net/weixin_47598409



【(11/29/180 5:14下午)(python@Sakura):~/桌面

0 (sys)

0 (0)

24 72 LOAD_NAME 7 (main) 74 CALL_FUNCTION 0 76 POP_TOP >> 78 LOAD_CONST 1 (None) 80 RETURN_VALUE

```
嵌入 payload 之后:
```

2

6 STORE NAME

8 LOAD CONST

11/29	9/18@ 5:31下午)(python@Sak	ura):~/桌面
pytho	on3 -m dis example.py	
1	O LOAD_CONST	0 (0)
	2 LOAD_CONST	1 (None)
	4 IMPORT_NAME	0 (sys)
	6 STORE_NAME	0 (sys)
	8 LOAD_CONST	0 (0)
	10 LOAD_CONST	1 (None)
	12 IMPORT_NAME	1 (os)
	14 STORE_NAME	1 (os)
		https://blog.csdn.net/weixin_4759
	80 KEIUKN_VALUE	
		•
E: Pa	ayload 的发送和接受方法完全取决于	用户个人喜好, Stegosaurus 只提供了一种向 Python
节码文	件嵌入或提取 Payload 的方法。但	是为了保证嵌入之后的代码文件大小不会发生变化,因此
Stegos	aurus 所支持嵌入的 Payload 字节	长度十分有限。因此,如果你需要嵌入——个很大的

Payload ,那么你可能要将其分散存储于多个字节码文件中了。

为了在不改变源文件大小的情况下向其嵌入 Payload ,我们需要识别出字节码中的无效空间(Dead Zone)。这里所谓的无效空间指的是那些即使被修改也不会改变原 Python 脚本正常行为的那些字节数据。

	1 NX+++, 1R1	<i>酒1</i> 采
作码是否带参,字节码中的操作指令将需要占用 1 个字节或 3 个字节。在 Python3.6	中就不一样了	٢,
Python3.6 中所有的指令都占用 2 个字节,并会将无参数指令的第二个字节设置为 0	,这个字节在	其
运行过程中将会被解释器忽略。这也就意味着,对于字节码中每一个不带参数的操作指令,	Stegosaurus	都
可以安全地嵌入长度为 1 个字节的 Payload 代码。		

我们可以通过 Stegosaurus 的 -vv 选项来查看 Payload 是如何嵌入到这些无效空间之中的:

#(11/29/18@10:35下午)(python@Sakura):~/桌面
 python3 -m stegosaurus example.py -s -p "ABCDE" -vv
2018-11-29 22:36:26,795 - stegosaurus - DEBUG - Validated args
2018-11-29 22:36:26,797 - stegosaurus - INFO - Compiled example.py as __pycac
2018-11-29 22:36:26,797 - stegosaurus - DEBUG - Read header and bytecode from
2018-11-29 22:36:26,798 - stegosaurus - DEBUG - POP_TOP (0)
2018-11-29 22:36:26,798 - stegosaurus - DEBUG - POP_TOP!(0)

python3	-m stegosaurus	example.py -s	-p "ABCDE" -vv
2018-11-29	22:36:26,795 -	stegosaurus -	DEBUG - Validated args
2018-11-29	22:36:26,797 -	stegosaurus -	<pre>INFO - Compiled example.py aspycac</pre>
2018-11-29	22:36:26,797 -	stegosaurus -	DEBUG - Read header and bytecode from
2018-11-29	22:36:26,798 -	stegosaurus -	DEBUG - POP_TOP (0)
2018-11-29	22:36:26,798 -	stegosaurus -	DEBUG - POP_TOP (0)
2018-11-29	22:36:26,798 -	stegosaurus -	DEBUG - POP_TOP (0)
2018-11-29	22:36:26,798 -	stegosaurus -	DEBUG - RETURN_VALUE (0)
2018-11-29	22:36:26,798 -	stegosaurus -	DEBUG - BINARY_SUBTRACT (0)
2018-11-29	22:36:26,798 -	stegosaurus -	DEBUG - RETURN_VALUE (0)
2018-11-29	22:36:26,798 -	stegosaurus -	DEBUG - BINARY_TRUE_DIVIDE (0)
2018-11-29	22:36:26,798 -	stegosaurus -	DEBUG - RETURN_VALUE (0)
2018-11-29	22:36:26,798 -	stegosaurus -	DEBUG - BINARY_MULTIPLY (0)
2018-11-29	22:36:26,798 -	stegosaurus -	DEBUG - RETURN_VALUE (0)
2018-11-29	22:36:26,798 -	stegosaurus -	DEBUG - BINARY_ADD (0)
2018-11-29	22:36:26,798 -	stegosaurus -	DEBUG - RETURN_VALUE (0)
2018-11-29	22:36:26,798 -	stegosaurus -	DEBUG - POP_TOP (0)
2018-11-29	22:36:26,798 -	stegosaurus -	DEBUG - RETURN_VALUE (0)
2018-11-29	22:36:26,798 -	stegosaurus -	INFO - Found ${\color{red} 14}$ bytes available for ${\color{red} r}$
Payload emb	edded in carrie	er	
2018-11-29	22:36:26,799 -	stegosaurus -	DEBUG - POP_TOP (65)A
2018-11-29	22:36:26,799 -	stegosaurus -	DEBUG - POP_TOP (66)B
2018-11-29	22:36:26,799 -	stegosaurus -	DEBUG - POP_TOP (67)C
2018-11-29	22:36:26,799 -	stegosaurus -	DEBUG - RETURN_VALUE (68)D
2018-11-29	22:36:26,799 -	stegosaurus -	DEBUG - BINARY_SUBTRACT (69)E
2018-11-29	22:36:26,799 -	stegosaurus -	DEBUG - RETURN_VALUE (0)
2018-11-29	22:36:26,799 -	stegosaurus -	DEBUG - BINARY_TRUE_DIVIDE (0)
2018-11-29	22:36:26,799 -	stegosaurus -	DEBUG - RETURN_VALUE (0)
2018-11-29	22:36:26,799 -	stegosaurus -	DEBUG - BINARY_MULTIPLY (0)
2018-11-29	22:36:26,799 -	stegosaurus -	DEBUG - RETURN_VALUE (0)
2018-11-29	22:36:26,799 -	stegosaurus -	DEBUG - BINARY_ADD.(@)dn.net/weixin_4759840

2018-11-29 22:36:26.798 - stegosaurus - DEBUG - RETURN VALUE (0)

```
2018-11-29 22:36:26,798 - stegosaurus - INFO - Found 14 bytes available for p
Payload embedded in carrier
2018-11-29 22:36:26,799 - stegosaurus - DEBUG - POP TOP (65) ----A
2018-11-29 22:36:26,799 - stegosaurus - DEBUG - POP_TOP (66) ----B
2018-11-29 22:36:26,799 - stegosaurus - DEBUG - POP_TOP (67) ----C
2018-11-29 22:36:26,799 - stegosaurus - DEBUG - RETURN_VALUE (68) ----D
2018-11-29 22:36:26,799 - stegosaurus - DEBUG - BINARY_SUBTRACT (69) ----E
2018-11-29 22:36:26,799 - stegosaurus - DEBUG - RETURN_VALUE (0)
2018-11-29 22:36:26,799 - stegosaurus - DEBUG - BINARY TRUE DIVIDE (0)
2018-11-29 22:36:26,799 - stegosaurus - DEBUG - RETURN_VALUE (0)
2018-11-29 22:36:26,799 - stegosaurus - DEBUG - BINARY_MULTIPLY (0)
2018-11-29 22:36:26,799 - stegosaurus - DEBUG - RETURN_VALUE (0)
2018-11-29 22:36:26,799 - stegosaurus - DEBUG - BINARY_ADD (0)
2018-11-29 22:36:26,799 - stegosaurus - DEBUG - RETURN_VALUE (0)
2018-11-29 22:36:26,799 - stegosaurus - DEBUG - POP_TOP (0)
2018-11-29 22:36:26,799 - stegosaurus - DEBUG - RETURN_VALUE (0)
2018-11-29 22:36:26,799 - stegosaurus - DEBUG - Creating new carrier file nam
2018-11-29 22:36:26,799 - stegosaurus - INFO - Wrote carrier file as __pycach
                                                                            .
```

参考文献

- <u>https://bitbucket.org/jherron/stegosaurus/src</u>
- <u>https://github.com/AngelKitty/stegosaurus</u>
- <u>https://www.freebuf.com/sectool/129357.html</u>

https://blog.csdn.net/weixin_47598409

Bugku-MISC- apple

(这题不会做,拿大佬WP)

apple 200

Apple在2017年WWDC上发布了全新的APFS,带有一系列新功能。 Bugku很好奇,马上试了一下。而且,他给你留下了一些惊喜:)

N1CTF_APFS_v			
Flag		S https://blog.csdn.ne	Submit atweixin_47598409
附件是一个dmg格式的文件,dmg是MacOS上的压缩镜	象文件,相当于在Window	ws上常见的iso文件	
名称 个	修改日期	类型	大小
N1CTF_APFS_ver2.dmg	2020/10/6 10:55	DMG 文件	8,344 KB
apple Apple在2017年WWDC上发布了全新的APFS,带有-	-系列新功能。 Bugku很妙	子奇,马上试了一下。 而且	1, 他给你留下了一
N1CTF_APFS_ver2.dmg 尝试打开发现有密码,使用010editor看到最后有个字符	符串:*		
88 E9 7A 5D FA 5A D7 75 D5 3A 0D 4F FD B6 A7 58 BD F3 3E 2B 20 E1 21 36 5F 41 50 46 53	0 0A 09 7A BA 8D 86 93 0F 41	<pre>™1SRI<ez] >,@à™Õ:.Oy à>°Y-½ó>+ N1CTF_APFS</ez] </pre>	uZ×uz ý¶§X°.† á!6".A S
使用N1CTF_APFS成功打开并挂载。打开挂载盘看到53 照:	31个0字节的txt,思路中题。	f查看writeup得知可以通	<u> </u>
<pre>1 #查看快照 2 tmutil listlocalsnapshots /Volumes/N1CTF_APFS 3 #挂载快照 4 mkdir temp 5 mount_apfs -s ctf /Volumes/N1CTF_APFS ./temp/</pre>	5		



22 for i in range(0, len(b), 8): 23 f.write(chr(int(b[i:i+8], 2)).encode('utf-8'))

写入之后用010editor打开发现是个zip,使用之间得解压密码N1CTF_APFS可得flag。*

N1CTF{APFS_a_N3xt_30_Year_Filesystem}

N1CTF{APFS_a_N3xt_30_Year_Filesystem}

Bugku-MISC-妹子的陌陌

Challenge

1308 Solves

х

妹子的陌陌 250

想要妹子陌陌号吗? 做题来拿吧

下载这个图片做题

momo.jpg

Flag



https://blog.csdn.net/weixin_47598409





https://blog.csdn.net/weixin_47598409

打开图片,这。。。。。。。不敢多想其他,还是赶紧把它做出来吧

<pre>[root@root~14:23:07~Tony] →~/桌面# binwalk momo.</pre>		momo.jp <mark>g</mark>		
DECIMAL	HEXADECIMAL	DESCRIPTION		
0 37340	0×0 0×91DC	JPEG image data, JFIF standard RAR archive data, version 4.x,	1.01 first volume type:	- MAIN_HEAD
Processing	ot~14:23:11~Tony] → ~/桌 面 # foremost : momo.jpg ot~14:23:20~Tony]	momo.jpg Hilfs I A Bugkova III M diriy Koy.tac.gz diriy ABIDJE	https://blog.csdn.net/we	aixin_47598409

用binwalk查看图片有一个rar文件并用foremost把它分离



居然有密码,爆破密码没啥用,然后仔细看了一下图片有一句话,试试这句话是不是密码,结果居然成功了。解压密码竟然是图 片中的"喜欢我吗."

> 00000072

名称		修改日期	类型	大小
momo.	txt	2017/2/11 22:50	文本文档	1 KB
momo.b	xt - 记事本			
文件(F) 编辑	晶(E) 格式(O) 查看(V) 帮助(H)		
嘟嘟嘟嘟				
士兵: 报告	皆长! 已截获纳料	的加密电报!		
首长: 拿来	活着			
电报内容:				
电报内容:	////./-	.////.///	/////	///
电报内容: /-/-/	////./-	.///.///	/////	///
电报内容: /-/-/, 首长:我摘	////./-	·.////.///	/////	///
电报内容: /-/-/ 首长: 我擁 士兵: 难違	////./- 桑你在逗我吗? 你碰 道我弄错了? 哦。。	·.////./// 触定是他们纳粹发的吗? 。等等是这一条	/////	///
电报内容: /-/-/, 首长:我摘 士兵:难道	////./- 操你在逗我吗?你碰 貸我弄错了?哦。。	·.////.///	/////	///
电报内容: /-/-/, 首长:我擁 士兵:难違 内容:http	////./- 桑你在逗我吗? 你确 值我弄错了? 哦。。 p://c.bugku.com/	·.////./// 確定是他们纳粹发的吗? 。 等等是 这一条 /U2FsdGVkX18tl8Yi7Fad	///// Giv6jK1SBxKD30e	eYb52onYe0=
电报内容: /-/-/, 首长: 我擁 士兵: 难違 内容: http AES K	////./- 操你在逗我吗?你确 算我弄错了?哦。。 p://c.bugku.com/ ey: @#@#¥%	·.////./// 靛定是他们纳粹发的吗? 。 等等是这一条 /U2FsdGVkX18tl8Yi7Fac ¥¥%%&¥	///// Giv6jK1SBxKD30e	/// eYb52onYe0=
电报内容: /-/-/ 首长: 我擁 士兵: 难違 内容: http AES K	////./- 操你在逗我吗?你确 直我弄错了?哦。。 p://c.bugku.com/ fey: @#@#¥%	·.////./// 全是他们纳粹发的吗? 。等等是这一条 /U2FsdGVkX18tl8Yi7Fad ¥¥%%&¥	///// Giv6jK1SBxKD30e	eYb52onYe0=
电报内容: /-/-/, 首长: 我擁 士兵: 难道 内容: http AES K 士兵: 二维	////./- 操你在逗我吗? 你確 值我弄错了? 哦。。 p://c.bugku.com/ fey: @#@#¥% 售码真的扫不出来吗	·.////./// 靛定是他们纳粹发的吗? 。等等是这一条 /U2FsdGVkX18tl8Yi7Fac ¥¥%%&¥ 3? ? 肯定可以扫出来	///// Giv6jK1SBxKD30e	/// eYb52onYe0=

里面有一个文本, 文本内容就是这样

/-/-/	-///	/.////.	////	////	.//
					Ξ
克莱斯密码	NEPTER	交換内容 洲	I空 下戰加密/	解密代码 复织	■ 刻如密/解密代码

发现有摩斯密码,直接解码,得出的字符好像是一条网址 HTTP://ENCODE.CHAHUO.COM/

← -	C ▲ 不安全	encode.chahuo.c	om							☆	* (8	1
III 应用	8 😤 百変 🔳 🔳	. 🔳 📲 🖷											
===	chahū0.com	AB abshu. com											
在线	加密解密 encode	& decode											
10180403	 字符串												
												1	
SHA	1 SHA224	SHA256 SHA3	84 SHA512	MD5 Hin	nacSHA1 Hma	CSHA224 HmacSH	HA256 HmacSH	IA384 Hn	hacSHA512	HmacMD5			
Unit	ncode UriDecod	AES加密	AESME	DES加密	DES解密 Rat	Rabbitie	RC4加密	RC4解密	TripleDESto	Triple(DESIR	2	
base	64tos base64	解密											
结果													
结果													

打开网站居然是一个在线加密解密网站, 收藏一波

把网址后面的base64作为密文粘进去,填上key,可解出明文: (注意:这个是AES解密)

在线加密	解密 enco	de & decode		
加密前字符串				
U2FsdGVk	<u> <18tl8Yi7Fa</u>	Giv6jK1SBxK	D30eYb52or	<u>1Ye0</u> =
密钥				
@#@#¥%	¥¥%%	&¥		
SHA1	SHA224	SHA256	SHA384	SHA512
UrlEncode	UrlDeo	code	AES加密	AES解密
TripleDES	解密	base64加密	base64	驿密
结果				
momoj2j.pn	g			

将解出的明文和网址前半部分网址合起来即http://c.bugku.com/momoj2j.png



使用stegsolve取反后用QR Research可扫出flag:



<	https://blog.csdn.net/weixin_47598409

😳 QR Research	_		×
文件(F) 工具(T) 帮助(H)			
● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	·级	掩码— Auto 尺寸— 4	
			^
			~
			^
已解码数据 1:			
位置:(6.7,4.0)-(697.4,4.2)-(6.7,693.4)-(699.4,693.5) 颜色正常,正像 版本: 2			



https://blog.csdn.net/weixin_47598409

KEY{nitmzhen6}

Bugku-MISC-就五层你能解开吗



就五层你能解开吗 300

链接: http://pan.baidu.com/s/1i4TQoz7 密码: w65m

- 提示: 第一层: CRC32 碰撞
- 第二层: 维吉尼亚密码
- 第三层: sha1碰撞
- 第四层: md5 相同文件不同
- 第五层: RSA

Flag

Submit

https://blog.csdn.net/weixin_47598409

下载附件

Challengs: Cryptography 500.7z 全接 下载(247KB) 말 保存到手机 举报 回生成链接


解压居然要密码



然后根据题目的指示

提示:第一层: CRC32 碰撞

第二层: 维吉尼亚密码

第三层: sha1 碰撞

第四层: md5 相同文件不同

第五层: RSA

先直接用CRC32碰撞脚本跑一波

脚本传送门https://github.com/theonlypwner/crc32

依次碰撞三个txt文件



verification	checkst	in: 0×/czur9	10 (UK)	
alternative:	5EJeBD	(OK)		
alternative:	74bFvQ	(OK)		
alternative:	D4WldU	(OK)		
alternative:	Jvea5S	(OK)		
alternative:	OSgAFe	(OK)		
alternative:	WtUlWB	(OK)		
alternative:	XgD1qA	(OK)		
alternative:	_3n26b	(OK)		
alternative:	_CRC32	(OK)		
alternative:	aSKHAn	(OK)		
alternative:	dvIh2X	(OK)		
alternative:	fJLvkE	(OK)		
alternative:	hESFWK	(OK)		
alternative:	l1r6Sx	(OK)		
alternative:	pbakFl	(OK)		
alternative:	uGcK5Z	(OK)		
alternative:	vgh8vJ	(OK)		
alternative:	xt8TKP	(OK)		
alternative:	ytyePI	(OK)		https://blog.csdn.net/weixin_47598409
		-	CARE AND PARTY	

-[root@root~18:39:28~Tony]

└ ── →•	~/桌面/d	rc32# pytho	n crc32.py	reverse	0×A58A1926
4 bytes: {0×a	ad, 0×d5	5, 0×fa, 0×7	8}		
verification	checksu	um: 0×a58a19	26 (OK)		
alternative:	1Jnhwi	(OK)			
alternative:	3W5fG8	(OK)			
alternative:	LEDrYc	(OK)			
alternative:	N4lQmv	(OK)			
alternative:	Tbv_HD	(OK)			
alternative:	ZmiotJ	(OK)			
alternative:	_i5_n0	(OK)			
alternative:	bxy760	(OK)			
alternative:	jS1DST	(OK)			
alternative:	kSpuHM	(OK)			
alternative:	lJwKbf	(OK)			
alternative:	rhL5Cg	(OK)			
alternative:	s9oe4b	(OK)			
alternative:	stBXYj	(OK)			
alternative:	tmEfsA	(OK)			
alternative:	zbZV00	(OK)	VSIIIIIII	nttps://blog.csdr	n.net/weixin_47598409

-[root@root~18:39:46~lony]	
	4DAD5967
4 bytes: {0×1b, 0×d6, 0×38, 0×c2}	
verification checksum: 0×4dad5967 (OK)	
alternative: 9rNYn3 (OK)	
alternative: Ay8sZC (OK)	
alternative: QHSaFX (OK)	
alternative: TmQA5n (OK)	

alternative:	VQT_ls	(OK)	test_data.py
alternative:	X28BT9	(OK)	
alternative:	_GLQzV	(OK)	
alternative:	goMEPt	(OK)	
alternative:	nyUKFQ	(OK)	
alternative:	t_s4f3	(OK)	
alternative:	xQxVkx	(OK)	
alternative:	yQ9gpa	(OK)	https://blog.csdn.net/weixin_47598409

从结果上猜测最有可能的组合是: _CRC32_i5_n0t_s4f3

使用这个密码成功解压7z,再解压里面的7z,看到一个keys.txt,里面有10000个密钥,一个ciphertext.txt,里面有一段密文。还 有个tips.txt:

> Challengs: Cryptography 500 > CRC32 Collision

ISKX HIMI	突尘	入小
2016/10/6 23:21	文本文档	1 KB
2016/10/8 1:59	7Z 压缩文件	3 KB
2016/10/7 2:25	文本文档	411 KB
2016/10/7 3:37	文本文档	1 KB
	2016/10/6 23:21 2016/10/8 1:59 2016/10/7 2:25 2016/10/7 3:37	2016/10/6 23:21文本文档2016/10/8 1:597Z 压缩文件2016/10/7 2:25文本文档2016/10/7 3:37文本文档

🥘 keys.txt - 记事本 文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H) ZVVOKMTHJNWFPFRLDWGWOPMDHMDDDMSHWJMNNKXZ ZSWKILXTWXOTAEISBTOBMRWBCODLSAOEHLKDGWDB KAMOYTOMBEUGOIMROKOTXPXUTOREHDHLFDIWCVPY ITBENEXUYTDZPXAWIWZIADYNMDGFEOSLCYMYXSKP QFLXKEBGTEFGAISNRPUJJHIORMICAFPDDQPKXVJG OBQPOELJUZFLCGFPAHEYISIILNKVIOCLQGLBFNYT XCVHPWOOZRELISPSTILVLLHRLUDFMMRSMCAUBVRD CJGVTUAQWHEYCIURHDCAOBVCDSVYTZRSPEFATZRG FNDTXXGRMYTFPPQAKKDIISDEHBQGMIAAMJTIRMAO NPKTNZDGJGEAHRVMANIJTZEBJVKWUAWTSKVBUFPQ NXQRVPJPBAMVTFJUTBFKSTYZJZDTNGXHCJPIHBDZ SHINMBHZYLJELSOHIBBWZMPOQTEWYFQQXPSNNIZW WMVSZTKFGGWAEGUMMKNLWRXBMGYFMZPJVFETHPTL DBTWAQAAAYVGTLILUEOVBNJUTSHHLVJRIAWJQLMW WAPAACVIGCBMNYIOLZNAKOEORLQFLNYDDJOVFKBG WVCJCTLXGKSEJBMFLXHAUTNNUTDLNWAPUDYOEOGR VYLNBTBDKDEXRBXGRKQFLRVBCZTYKDGLVMCITQJP LSOAMNTLIOLRRROIWHTAXVVNCKQAKFWGAJKEHPOL XMFFUYVCRQHXEVBNRTFJHQAOGEDPTIEVTMITBNSP BHSBMCFLHBUKSRDLZPMDQPTFKCECDJUOFHREDMWX PDTPCQCDVFFTGRDDWQNEYQWGRWODHPEUKXIIYKMF GWUDVMWGNAQGHEKVMHFEKFZPSGMRBRHWISKLZFRE SFTUUBRUPFLLEZANYORAFYLDEFBNNXCYKZPXNNSO

🥘 ciphertext.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

rla xymijgpf ppsoto wq u nncwel ff tfqlgnxwzz sgnlwduzmy vcyg ib bhfbe u tnaxua ff satzmpibf vszqen eyvlatq cnzhk dk hfy mnc 🗠

📃 tips.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

你知道维吉尼亚密码吗?

我们给了keys.txt, 唯一的密钥就在其中, 那么解密ciphertext.txt里的密文吧! 解压密码就在明文里, 祝你好运!

Do you know the Vigenére Ciphers? We gave the keys.txt, Only have a key in it, So decrypts ciphertext.txt! Unzip Password in plaintext, good luck to you!

https://blog.csdn.net/weixin_47598409

写个python批量解密维吉尼亚密码:

from pycipher import Vigenere
keys = []
cipher = 'rla xymijgpf ppsoto wq u nncwel ff tfqlgnxwzz sgnlwduzmy vcyg ib bhfbe u tnaxua
ff satzmpibf vszqen eyvlatq cnzhk dk hfy mnciuzj ou s yygusfp bl dq e okcvpa hmsz vi
wdimyfqqjqubzc hmpmbgxifbgi qs lciyaktb jf clntkspy drywuz wucfm'
with open('./keys.txt', 'r')as f:
 keys = f.readlines()
plain = ''
with open('./plain.txt', 'w')as f:
 for i in keys:
 i=i.replace('\n','')
 plain = Vigenere(i).decipher(cipher)
 f.write(plain+'\n')

/root/桌面/Challengs	Cryptography 500/CRC	32 Collision/plain.txt - Mouse	epad	- • ×
文件(F) 编辑(E) 搜索(S) 视图(V) ;	ζ档(D) 帮助(H)			
警告:您正	在使用 root 账户,搁	作不当可能会损害您的	系统。	
警告:您正 \$QFJOAPCXCJKANXILAKYZYQTXZCCQTYEM STENQBLQKSRWPOGBNDCTBWGVCXCUBFCHZ HLOHATUXFLLJBKCCAMCBQYFCLXOBMCJAE JSZTLILPIWCQAVOXGARMNKEJSIZAPRYAE BGPAOIHDNLAJPKWGXHWLEGUINZXDTABII DKKIKIXAMQAENMJEOPMWFVUOTYVKLROAC UJFQJQUTHYBEHAZBVOFZCCVFTRCAHTZTU PCUCFSITKIBRNKUCHTOUZMHUBTKHAGZTE MYXEBPCSURMKADYTEMNMFVZSXKPZHXQLU EWQELNFDXJBPIBTHOJILUOYVVQVJZFUSC EOKGDXZUFPTUWNFZVVLKVUEXVMCMGZTEE ZESKMLBKIEWLEAAMGVPYOBNIOSBJVAAV VZFFZTYEAJJPLMUHCMDJRWFVSFHAHGBCU OKHBYWIJGRKJWHGIUSCZMATCLTYYIKHUY VLLXYKNBANEDCUGFDXDUDZYINAPAISSII VQYOWTXMAFNLGRCODZJUTUPJKSCUGJQWM	在1使用 root 账 ア , 発 ELJMPVHOQBPKNTRM CNTTDPFOTATMOPFZI KPAXEDIDLKNWGYMU PLYCHIRFSVZHVPEX: XIMCEJAXGLGKVLSC HMVUMUNZKGOGVBPBI LXCYEBDQJBWFDYIW JSWGABALCQIBFMIZ EEOINSSIYTKXCGHJ DCVFUDQAWMKHAJSM EIOSYPHAOGMZKDJUC YFJMRTKVEOQIYFZXI ITDSKZVRZBLVGCTPW NBNJOGKKKDHUJMYVI EJBUPRARLHDUXRQP/ KZIVLMPRQUUSGBP	除作不当可能会损害您的。 PCBTWKWCFYYFYOUNOCCP FKNIXTPEIQTHWEWSMCUA (FAUTPQRRQBWYDDBMOBL JVHTEBLXFFMTKCKIXZAO (TAITJUOMKLKGSJDOXDS DTVGVWSFUAWLVSPJNVKK LUVAJMPMTTZICTGJGCAG /UIGTHQYYCUFMFVRIKHZ EFBTMLHVRBMLVXTNZPZG WUGBKGVFOWLAOWWLFVJY CMLPWSNMAZKBUCYLBCMF RPCXJNAXADYUBLJEHBJU WCGEVHVTRRJXWDWIUHAG EDAPQTWLXQZSARDBIYYH AXUVDTTUCRUJZWJDPPAH SGCZAPCUEXUZUNKAHCUF	祭 邦売。 OBSQGWOSLVARJSQOJSTKDC AFVFEYYZZPYRMRUQKOHXTC XMOHMAFDADBGEBOACXOSMJ MBOKRWDIDIKILMALROGVXZ VEWJZTRIAJTAZCHORKUAM) MROWJXAAIUGCDXPKRARZR) OCHANIHEABWTCSXJZXKUZY BCHXLDBMWBTOVHJFBLKXJY STZAGPTOJSLLRKLBYFJHS: AXGUFNALQDJDPDLLWIUKKY UWSKGTTYUPADHXNDGCLSQC VDJPAQOSNTDYXFRMUEBVFJ BEQRKEIYGZOUSSMZCBVNKC FKIEPMSPKGDNDUIYFLATSF NVWJGUGALRTUEYEYTQSNOG	GIKCTSHI DUZDCLJ((HLZYMW) ZOKKHCI (HZZSQTS (CXBFOK/ (CRPVLR2 (PXZQMDE DIKSUDA) (NSQPRKI QSGCBJR(FJOPWWC(SNVBQRY) HGWCSQI BBMJCPZI (JQGYYZ)
WNPKXTHGWMBSYRRNXMAPCWHVCMMHJCKAL GTMXMZPYYBUYYBALSPXUQSHJCBPFJAUF(BVOGJJRSNLQTGLVL2 ENSSKEVCTIDIMTNN(ZUJRAEBOROPIWFWSBMHI DNPRKNZJULUWSFKSQPFI	XNORDGTMFJKVGCRXCKXJN\ VXTMGYXYGEOFMZEMISPLCF	QIGNXT)
			SPETDAGETAKTSIZEXQYGAX	

从得到的10000个明文中,搜索"Vigenere",找到唯一一条有意义的明文:

T F

MIYBN THEVI ELSII	IKVESATVVO GENERECIP KUHYVTRMH	NOOHEDHASLCQRMDA HERISAMETHODOFEN HKXQQVOCDCAVPSBF	ZBETEERZPSIIUHJXZTH ICRYPTINGALPHABETICT YXPJDNKQVSTALOOEXYWN	INMMOXACFMSDNA EXTBYUSINGASE IIMIFOUYOVEVUC	QMSFRMCVCPKVPJ) RIESOFDIFFEREN BDQKPSAKADPFYPI	XVPLLBLINTYVFBQTFDJYK TCAESARCIPHERSBASEDON WRSMDEFPITXYLWQPWWDUT
×	查找(N):	Vigenere	↓ 下一个	↑ 上一个	Ⅲ 全部高亮	区分大小写(C)
IGENEREC	CIPHERISAM	ETHODOFENCRYPTIN	GALPHABETICTEXTBYUS FAKEYWORDTTTSASTMPLE	INGASERIESOFDI FORMOFPOLYAL	F	
ETICSUBS	STITUTIONS	OPASSWORDISVIGEN	ERECIPHERFUNNY			

从最后可以得到下一个压缩包的密码是"vigenere cipher funny",需要全部改为小写。

Challengs: Cryptography 500 > CRC32 Collision > Find password

^	名称	修改日期	类型
	Easy SHA1.7z	2016/10/8 1:59	7Z 压缩3
	U need unzip password.txt	2016/10/7 3:26	文本文档

https://blog.csdn.net/weixin_47598409

解压得到一个加密7z和一个txt:

🥘 U need unzip password.txt - 记事本	
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)	
现在我们遇到一个问题,我们有一个zip文件,但我们不知道完整的解压密码。 幸好我们知道解压密码的一部分sha1值。 你能帮我们找到的密码吗?	
不完整的密码: "*7*5-*4*3?" *代表可打印字符	
不完整的sha1:"619c20c*a4de755*9be9a8b*b7cbfa5*e8b4365*" *代:	表可打印字符
人生苦短,我用Python。	
Congratulations!	
Now we run into a problem,We have a zip file, but we don't know t Fortunately, we know that part of the unzip password of sha1 value can you help us to find the password?	the complete unzip password. e.
Incomplete password is "*7*5-*4*3?" * in the range of ASCII printa	ble characters
Incomplete sha1 is "619c20c*a4de755*9be9a8b*b7cbfa5*e8b4365*"	* in the range of ASCII printable characters
Life is short, you need Python.	

写个python跑一下:

import string	
import hashlib	
printable = string.printable	
for k1 in printable:	
for k2 in printable:	
for k3 in printable:	
for k4 in printable:	
key = k1 + '7' + k2 + '5-' + k3 + '4' + k4 + '3?'	
<pre>sha1 = hashlib.sha1(key.encode('utf-8'))</pre>	
<pre>flag = sha1.hexdigest()</pre>	
if '619c20c'and'a4de755'and'9be9a8b'and'b7cbfa5'and'e8b4365' in flag: print(key)	

挨个试跑出来的结果:

```
[root@root~19:21:38~Tony]

~/#面/Challengs:Cryptography 500/CRC32 Collision/Find password# python q.py

8725-{4}3?

3?B5-]4

k7T5-)4l3?

p7.5-"4'3?

F7M5-i4@3?

I7~5-s4F3?

*705-q4w3?
```

逐个测试最后发现"I7~5-s4F3?"是真正的密码。解压后看到一个7z和一个txt:

/root/J	東面/Challen	gs : Crypto	graphy 500	/CRC32 Col	lision/Find p	assword/Easy SH/
文件(E)	编辑(E)	搜索(<u>S</u>)	视图(⊻)	文档(D)	帮助(出)	
			警告:您	正在使用「	root账户,	, 操作不当可能
Hello Wo MD5校验真 有没有两个 如果有的语 解压密码为	rld;-) 的安全吗? 不同的程序 另一个程序 单行输出组	RMD5却相同 R输出是什么 結果。	司呢? 么呢?			
Hello Wo MD5 chec There ar If so wh The deco	orld ;-) k is rea e two di at is th ompressio	ally safe ifferent ne outpu on passwo	e? procedu: t of ano ord is a	res MD5 ther pro single-	is the sa gram? line out _l https://blog.csdr	ame? put. n.net/weixin_47598409

然后从网上找到了这样的两个软件:



输出分别是:

C:\Users\Administrator\Desktop\qq>HelloWorld-colliding.exe Hello World ;-)

C:\Users\Administrator\Desktop\qq>

C:5.	管理员	ដី: C:\Win	dows\system32\cmd.exe - GoodbyeWorld-colliding.exe
Good	dbye	World	:-(
Good	ibye Hhvo	World World	:-(·-(
Good	dbye dbye	World	:-(
Coor	lhiro	Womld	

经尝试 Goodbye World :-(是正确的密码。

这里涉及MD5的一个漏洞知识,可以自己科普一下——MD5碰撞生成器

解压之后得到一个pem密钥文件,和一个enc密文文件。



下载RSA攻击脚本 传送门https://github.com/Ganapati/RsaCtfTool

安装脚本的步骤: git clone https://github.com/Ganapati/RsaCtfTool.git sudo apt-get install libgmp3-dev libmpc-dev pip3 install -r "requirements.txt" python3 RsaCtfTool.py

运行以下命令:

python3 RsaCtfTool.py --publickey './rsa_public_key.pem' --uncipherfile './flag.enc'



flag{W0rld_Of_Crypt0gr@phy}



<u>创作打卡挑战赛</u> 赢取流量/现金/CSDN周边激励大奖