RSCTF一些WP (完全小白的答案)

ttttimer? ① 于 2019-10-31 21:33:02 发布 454 🏫 收藏 1
t章标签: <u>CTF</u>
反权声明: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。
文链接: <u>https://blog.csdn.net/Tttttttimer/article/details/102846119</u>
反权

1.签到题

Challenge	98 Solves									
车票拿好										
	1	LO								
Welcome to RS 号: NEEPU Sec	CTF! 各位小(&& 七色堇安全	火伴们, 想上车请 :	美 注微信公众							
Flag			Submit							
		https://blog.	.csdn.net/Tttttttime							
题面是这样的。很重 关注了七色堇安全斥	直白。 后会得到这样的:	消息								
18:21		17.3K/s 🕸 🗇 .	attl 4G# attl (



可以发现有个}于是应该是后半部分。

关注了NEEPU Sec虽然没有自动回复不过问一问对方就知道了。





于是得到了完整的flag{N33pU@CcuT}

2.国学

Challenge	Challenge 33 Solves											
国学												
	173											
玄学才是经典 艮坎震震艮坎5 兑艮艮坤坎坎5	玄学才是经典 题目:兑巽艮震震乾艮乾坎巽坤震艮艮艮震坎 艮坎震震艮坎坤坎坤乾震坎离艮离坎震坤震离离坤兑坎震坤震 兑艮艮坤坎坎坤震兑乾离乾坎坤坎											
	View Hint											
Flag		Submit										

https://blog.csdn.net/ I ttttttimer

题目给了一堆先天八卦,恰好我对玄学有所涉猎。八卦从乾到坤分别对应了一个数字。最大是八所以在这里应该是八进制。但是 八进制并没有8,所以所有的数字都减一,变成0-7的八进制。这里给出对应关系乾0、兑1、离2、震3、巽4、坎5、艮6、坤7。 (一开始忘记-1了于是思路卡在这里了,耽误了好久)

把原题里的放到word里,挨个替换。

146330605473666356533657570352625373227153731667557310205	75

🛛 查找和替	换		×
查找(<u>D</u>)	替换(P)	定位(G)	
查找内容()	l): 📃		~

选项:	区分全/半角
替换为(]):	6 ~
高级搜索()	<u>M</u>) * 格式(<u>O</u>) ▼ 特殊格式(<u>E</u>) ▼
替换(<u>R</u>)	全部替换(A)
	查找上一处(<u>B</u>) _ <u></u> , <u></u> <u></u> <u></u> , <u></u> <u></u> <u></u> <u></u> , <u></u> <u></u> <u></u> <u></u> , <u></u> <u></u> <u></u> <u></u> <u></u> <u></u> , <u></u> <u></u> <u></u> <u></u> <u></u> , <u></u> <u></u> <u></u> <u></u> <u></u> <u></u> , <u></u> <u></u> <u></u> <u></u> <u></u> <u></u> <u></u> <u></u> <u></u> , <u></u>

将得到的这串数字放到进制转换工具中进行8进制到16进制转换。

14633060	5473666356533657570352625373227153731(转换
进制	结果
2	110011001101100011000010110011101110110
8	1463306054736663565336575703526253732271537310
10	3832112901064781622774484826289738032467516065
16	666c61677b67756f5f7875655f69735f676f6#247d dn.nel/Tttttttimer

然后再把这串16进制数放到HEX里面就可以得到flag了~

₩ HxD - [尢标题1]																	
📓 文件(F) 編輯(E) 搜索(S) 查看(V) 分析(A) 附加(X) 窗口(W) 关于(A)																	
🗋 🚵 🗸 💭 🥔 🦉 💀 16 🔍 ANSI 🔍 十大进制 🗸																	
記 无标题1	國 无标题1																
Offset(h)	00	01	02	03	04	05	06	07	08	09	oa	0B	0C	0D	0E	OF	
00000000	66	6C	61	67	7B	67	75	6F	5F	78	75	65	5F	69	73	5F	flag(guo_xue_is_
00000010	67	6F	6F	64	21	7D											good!}



https://blog.csdn.net/Tttttttimer

3.XOR

这个xor.txt打开是JiwhJzshc3lyeXZwdHJxeXFzcXZ1cXl3eHR1JnN0liZ1JHEheD0= 最后有个=一看就是base64加密。于是先把它解一次base64。 解码后是这样的。

请将要加密或解密的内容复制到以下区域

&,!';!syryvptrqyqsqvuqywxtu&st"&u\$q!x=

然后这个题目一看肯定要用到异或的。搜了一圈也没找到在线工具,只能自己写了一个。



不知道key所以穷举出来,还好找到了flag{}这种格式的。

4.Pyc

Challenge	37 Solves	>	<					
	рус 10							
CrackMe								
📩 1.рус								
Flag		://blog.csdn.net/Tttttti	mer					
		所有 开发类	站长类	极客类	HR	其它	码农文库	奇淫巧技
		请选择pyc文件进行 选择文件 未选择(#!/usr/bin/env p # encoding: utf- # 如果觉得不错,可!	解密。支持所有 壬何文件 ython 8 以推荐给你的朋	『Python版本 朋友!http:/	//tool.	lu/pyc		
		def encode(str): s = '' for i in str x = ord(x = x + s += chr	: i) ^ 32 16 (x)					
		return base6	4.b64encod	e(s)				
		correct = 'KVEKK flag = '' flag = raw_input if encode(flag) print 'yes' else: print 'no'	FRUUVQijiU ('Input fl: == correct	mIyglvlnvj ag:') :	shRViQ	nusevviu https://b	umu1Y=' log.csdn.net	

首先这个pyc需要反编译,可以得到以下内容。

可以看到这里首先每一位都与32做异或,然后在+16之后进行base64加密。 只要反着按照步骤解回来就可以了。先进行base64解码,然后-16,最后再跟32异或。 最后得到flag:

Literation (TWT)4	
<pre>1 import base64 2 = "'VVMSRTHUTVijihmIyg[VINVjSktViGedJSHVVithHIT"' 4 = c.decode('base64') 5 = *' 6 for i in m: 7</pre>	9a48ddad2656385fce58af47a0ef56cf

5.听

Challenge	49 Solves	×								
听										
10										
flag提交格式为	g:flag{xxxx}xxx即为所得字符串									
🛓 lalala.gif										
Flag		Submit								

ttps://blog.csdn.net/Tttttttimer

这道题浪费了我好多时间。因为是这个图片是gif的,我就没一开始就先放进hex看看。习惯性打开了Stegsolve,但是没什么特别



把这个HEX的内容放进去可以得到一串摩斯密码。

₩ HxD - [尤标题1]																
盈 文件(F) 编辑(E) 搜索(S) 查看(V) 分析(A) 附加(X) 窗口(W) 关于(A)																
📄 🚵 🗸 💭 🥔 😃 💽 16 🗸 ANSI 🗸 十六进制 🗸																
記 无标题1	🔝 1.p	ус														
Offset(h)	00	01	02	03 04	05	06	07	08	09	0 A	0B	0C	OD	0E	OF	
00000000	2D	2E :	2E 3	20 2E	2E	20	2D	2E	2E	20	2E	2E	20	2E	2E	
00000010	2D	2D :	2E 3	2D 20	2D	2E	2E	20	2E	2E	20	2D	2E	2E	20	
00000020	2E	2E	20 3	2E 2E	2D	2D	2E	2D	20	2D	2E	2E	20	2E	2D	
00000030	20	2D	2E :	2D 2E	2D	2D										[]

跟这个Ascii里面的是一致的。

对不起	,请在两个连续莫斯密	闷中间加上空格.	1,
	↓word2morse	↑morse2word]
 —	—		
			/Tittitiigne

然后放到在线工具提示需要在两个连续的之间放上空格。

于是在这里卡了很久,换了很多个工具。幸好有个工具把不能翻译的部分完整的留下来了。

字母 Alphabet:
DIDIDIDIDA
摩斯密碼:

是中间那几个很长的,不是数字也不是字母,所以怀疑是符号。于是找了一个符号摩斯密码的对照表。

感叹号 (!)	
连字号 (-)	
下划线 (_)	
双引号 (")	

所以不能被工具翻译的内容就是下划线和感叹号了。最后把这些内容放进题目的flag格式里面,这次的比赛公告说基本都是flag{} 这种格式,所以就把字母都换成小写,于是就对了。(我后来想试试大写对不对,但是已经不能提交了。毕竟摩斯密码是不分大小写的)

6.veryeasypwn

flag提交格式

Web题目: flag在根目录带有flag关键字的文件中或web根目录下

首先我们要知道这个。

Challenge	16 Solves		×			
	veryeas 81	sypwn 9				
🛓 pwn2						
	Instanc	e Info				
	Remaining Tir	ne: 3595s				
	117.139.247.14:9754					
Destroy this instance Renew this instance https://blog.csdn.net/Ttttttim						

题面是这样的。

用nc连接进去发现似乎是个计算器。但是直接就报错退出了。



要想办法进入它的根目录才行,所以把下载下来的pwn2放到IDA里面可以看到它的主函数。

```
printf("Enter your choice:");
__isoc99_scanf("%d", &v4);
if ( v4 == 3 )
{
  fun_mul();
}
else if (v4 > 3)
{
  if ( v4 == 4 )
  {
    fun_div();
  }
  else
  {
    if ( v4 != 233 )
     goto LABEL_14;
    fun_sys();
```

```
}
 }
 else if ( v4 == 1 )
 {
  fun_add();
 }
 else
 {
   if ( v4 != 2 )
ABEL_14:
     puts("You've made some mistakes!!");
    return 0;
   3
   fun_sub();
 }
 return 0;
```

可以看到除了计算器有的内容外,额外还有一个调用了fun_sys();函数的入口,入口的密码是233。于是返回连接中输入233。



Cat一下这个flag就可以了。

7.Simple

Challenge	32 Solves	×
	Simple	
	224	
CrackMe		
🛓 simple.exe		
Flag		Submit

下载下来这个exe运行。随便输入一个数字就退出了。

💽 C:\U	C:\Users\Charon\Desktop\simple.exe									
lease	input	the	number	which	you	like:				

放到IDA里反汇编,居然没有主函数。没关系直接搜索flag。

					and the second se	
text:000000014000140E	sub_1400013AD		lea	rcx, aFlagS		"flag{%s}"
.text:00000001400016B2	sub_140001630		nov	ecx, eax	1	Flag
.rdata:0000000140006***		; Flags 400000	40: Dat	a Readable		
.rdata:0000000140006***		aFlagS	db í £	lag{%s}',0		DATA XREF:
.rdata:0000000140006***			dd O		-	GlobalFlag
rdata:0000000140006			44 O			GlobalFlag

这里有两个flag{%s}是我们要的格式,挨个点进去跳转看看。选择这个有函数名字的。查看它的代码。



于是可以看到输入的数字是233333时可以看到flag{%s},但是他会自己exit,所以需要一个断点让他停下来。 放到dbug工具里面,依旧搜索flag{%s}找到位置,为避免消失我在附近设了好多个断点。



在这些断点之间慢慢运行,可以观察到右边一些寄存器信息,于是就得到了flag。

•	隐藏F	PU	
	RAX RBX RCX RDX RBP	00000000000000000000000000000000000000	&"C:\\Users\\Charon\\Desktop\\simple.exe" "flag{%s}" "9cb18aef59e8f00877229bc75491f175"

8.RSA

这道题完全不会,但是我觉得一定有迹可循,于是我去i春秋的密码题里面搜了RSA,果然让我搜到了。

https://www.ichunqiu.com/writeup/detail/581

几乎完全一样,除了我们这个需要输入一串hex来换取flag而他直接丢进hex里面就可以了。

Interventerv

于是按照他的思路和脚本一步一步走下来最后就能得到结果了。

9.WF

首先把它运行一下看看,结果得到这样的运行结果。



可以知道它的主函数一定可以通过搜索Please找到。放到IDA里面寻找它的主函数。然后F5看它的伪代码,得到如下内容。



可以看到如果输入不是数字2的话将会显示byebye然后直接退出程序。输入数字2可以往下运行。下面的内容里有一句明显的"get the flag",也就是说这句显示之前就可以得到结果了。

而这句显示前有几个指令,一个是WF,一个是DF。写入然后直接删除。那么我们的任务就是让它停在这中间,找到写入的flag 文件。

放进debug工具,查找这句话的位置,在它前面设置断点。

	00007FF735673223	48:8BCB	mov rcx.rbx	1
	00007FF735673226	FF15 DC7D0000	call gword ptr ds: [<&WriteFile>]	
•	00007FF73567322C	48:8BCB	mov rcx,rbx	
•	00007FF73567322F	FF15 DB7D0000	call gword ptr ds: [<&CloseHandle>]	
•	00007FF735673235	48:8D0D CC490000	lea rcx, gword ptr ds: [7FF735677C08]	0
	00007FF73567323C	FF15 5E7E0000	call gword ptr ds: [<&DeleteFileA>]	
	00007FF735673242	48:880D 077F0000	mov rcx, qword ptr ds: [<@?cout@std@@3V?\$basic_ostream@DU?\$c	n
	00007FF735673249	48:8D15 98480000	lea rdx, gword ptr ds: [7FF735677DE8]	0
	0.00077777777737707	E O E O DE E E E E E		

然后F9运行,输入数字2调试。 然后在右边寄存器里其实flag就已经出现了。

朝田	A 12HS 13	12 · 10.0 · 17.07	日期 副規題技 📆 228社 三日日本 💾 行ち いまれが	2.8	H 70	E 🗑 Deswend	の資本語 🛋 河路 👔 第四年
*• 1	NAMES OF A DESCRIPTION OF	FF18 DC7D0000	[call gword ptr dos[-awriterile>]	~	REPORT		
•	000000000000000000000000000000000000000	48;8808	MOV FER, FDX		100000		
	00000755785478225	FF15 D87D0000	call gword gtr ds:[-odCloseHandle>]		0.4X 0.0		
•	0000777736473236	48:8000 CC490000	Tes rcs. geord ptr ds: [744736477004]	00	1000		
	100000766735673230	FF15 5E7E0000	call gyprd ptr doc[odDeletefiless]		585 10		
	0000788735673242	45:5500 07700000	more than parted att doi 1 and routdot and 107 than to out reading the		ACX 00	0000033999000035	
	0000777735673249	4515015 95450000	les out destrict str die TEFTIGATOFET	100	RDX 00	0000978786F768	"\$66c37703f08cc4913e30314e9075467"
- 11	0000788738478380	ER SCHEFFER	AND A TRAVESTICAS	1.00	R8P 00	000000000000000	
- 21	00000111110001010100	LO PROLITIT			8.55 00	00009782588720	
	000077777355573255	10109-0	many reneration		4/7 10	000000000000000000000000000000000000000	
	0000777735873258	4818015 OPDEPPPP	Tea rds. geord per del [PPP/sseraces]		8.51 00	000000000000000000000000000000000000000	
	00007FF73567325F	FF18 C8780000	call goord ptr ds: [sarrs tibasic_istreamouricher_traitum		8DI 00	DOUTH RECORDER	A"##ENTSLOGEEE1232+82965952"

原本我以为这是一个hex十六进制,但是是866c而不是666c所以在外面套上flag{}就提交看看,结果对了。

10.Alice is loney (为啥不是lonely)

因为之前的题目知道了RSA的主要格式,所以nc进看到内容的是就知道是RSA了。这里应该是已经c、n、e求m的算法。

这道题参考了王大锤的二哥王二狗的笔记的第三部分已知 公钥(n,e) 和 密文 c 求 明文 m?

https://blog.csdn.net/vhkjhwbs/article/details/101160822

但是这道题的数给的很有趣,把n拿去因数分解居然本身就是一个质数,因数只有她自己和1。王二狗给的脚本里q和p最小要是 2,所以王二狗的脚本就不能再用了。但是这次的e很有趣,是2。也就是说在该题设下,c=m的平方再对n取模。

很明显c的值远远小于n,那么在已知n有300多位的情况下,且n是一个大素数。m的平方只有两种情况,一种是m²=n+c,或者m² <n。

第一种情况太长了,所以先求一下第二种,当m²<n时,这个mod运算相当于没做。此时的m = gmpy2.isqrt©。 将这个m输出成hex,再输入到控制台里就出了flag。(果然没有太长)



11.饿了么

首先拿到一个普通的压缩包,里面只有个叫base的图片,没什么特别的。但是这个名字base却不叫肯德基或者KFC首先想到是

1895	6350	日時位	T1640							
iod.zi	p								-	当前
+ :	當称		大小	压缩后大小	美型	安全	像故时间	CRC32	压缩算法	38-
	(上层目录)									
🖬 b	ase.jpg		45.65 KB	36.40 KB	JPG 文件		2019-09-21 23:52:	8C30232C	Deflate	

base64加密。然后就没什么特别的内容了。

把图片解压出来放到HEX里面查看,发现冗杂的信息很多,而且我注意到了zip的文件头。可以看到里面隐藏了一个file.txt

 11C0
 08
 49
 80
 24
 84
 28
 92
 22
 50
 84
 24
 30
 42
 10
 90
 90
 90
 91
 .1€\$, ('"P,\$0B...

 11D0
 05
 08
 48
 03
 A2
 10
 84
 C6
 08
 42
 10
 00
 84
 21
 30
 04
 ..H.e., E.B.., !0.

 11E0
 D0
 84
 00
 90
 84
 20
 01
 08
 42
 10
 30
 42
 10
 D,..., E.B..B.OB.

 11F0
 80
 3F
 F
 D
 50
 4B
 03
 04
 04
 08
 02
 10
 D,..., E.B.OB.

 1200
 35
 4F
 F
 D9
 50
 4B
 03
 04
 04
 00
 00
 08
 00
 50..., E.G., \$w.....

 1210
 00
 06
 69
 6C
 52
 27
 78<</td>
 74
 4B
 57
 BA
 2C
 99
 53
 ...file.txtKN*, \$w\$
 \$w\$
 \$w\$
 \$w\$

于是把base.jpg变成base.zip得到一个加密的压缩包。

标记		
备注	I42FSRCNJVFFQR2NGNKEOTS2LBDV	
李 酒	URCEJ5GVEV2HKE6T2PJ5HU6Q====	
作去		
1F1=		
拍摄日期		

密码找了好久都没找到。。连CRC碰撞都快尝试了。迫不得已又看了一下图片详情~~(其实我刚拿到图片的时候就看了一次 的,可能当时没仔细看)~~

果然发现备注里有以====结尾的一串字符。

虽然一开始想到用base64,但是真的拿去解码的时候是一串乱码不能再继续了。而且这也是我第一次见到四个等号结束的base加密。就回到了开始的疑问为什么文件名是base而不是base64呢。于是我猜base除了base64可能还有别的方式。

于是找到了一个博客。这里参考。

https://www.cnblogs.com/superking-sukai/p/8491349.html

发现三个以上等号结尾应该是base32。于是去解码。这里需要解码两次,会得到16进制串。

G4YDMM_IXGM3TGNZXGZDDOMRWGQ======

编码解码

70617373776F7264

https://blog.csdn.net/Tttttttimer

丢进HEX得到password(就是password,直白的密码)。

解压缩得到file.txt,里面的字符也太多了。同样参考上面的博文,拿去做词频统计。(他的工具没用明白)用了一个简单的词频统计。

◎ 字词频率统计				
文字内容 (最长100000字) : 共 30500 字符		序 _{会谈}	5 52 1	-
ibAHfiefi3f2Hfi6ec(fi4gbfA2alefifagle3i2fF6se)kig34g1H2)caka6Fk421fc1(b11bA)			X	
(b (sF6tbgillf (2) sclbefe (A3kekHlbfge (364g (6fk (aflg (k (keF) 2fc44kkgaggibf22Hsifs) fa		1 {	1750 13. 2989	9
F86642+5220061143826 [8 (019)	字顶统计	2 }	1050 7. 9793	
From Diagonine ((CI))		3.6	958 7.2002	
ifaf (bfsff24gsslAA 90bf21c3fsfkb () fHeFb) 2 (lff1s2H21) g1 (ei4 (lsf4bs2 (e4f2afbafbfba4	一六黄风子	4.4	928 7.0522	
sclafbig) 11eg23cAHgsa (ss3ffA GbH6e2ibAiFfe16iFF2cH) 4iAAbifgkHA (gkAll2e3Fafieekkf		5 3	094 6.7930	
ffH3Acfg16fHikg24g26k3Akcg2g21cAi14{iFfH3kk1H6fcii14i6iikHk634gg3a3a3aff621F31gg2		6 2	864 6.5658	
fang Th 2 [an] 22 [ant for any h] Bo [for a h Of] 6h 2 df (HA f) of f GB fHa; f dd; do Wild Fo [ant GHo of 2422	间频统计	T f	258 1.9606	
a big of the off the first the second of the second s		8 1	152 1.1551	
1c1 (1 (a%kfck) HgfA (f20HAbgHbebgfbabAFgb10kefHbF (311F2F310gfb4aA31c (a1fFH (fbeF (4kc		2 a	142 1.0791	
k (Pbagfaklkgk4f) Ak2Asbk4kfkbgsb3ia { ffPi43AH { [4HgAia6Hkf3 {4iF {gsPsbffHb3ai61Ffffi		10 b	123 0.9347	
kf) 1a46ffgfa (12cf64Hb23fFffH6f4cs4ia) FFf4ekbbf3 (2 (gHbAg (i fbifa6aA4e4ck (ffFc2 (1s6		11 g	123 0.9347	
15 + 1 12 at of Falls backet debt and deliber (Will at (at ant oth) 175 123 + 136 + 02 + 06. Second) W1 234 + 24		12 H	120 0.9119	
sector is a sector base of the sector of the sector of the sector of the first sector of the first sector of the s		13 i	111 0.8435	
goffHHclisek2Abstalf)g (logiFs3ak3Hafaf) llog3HeclgaHAgaffiolle2ellabf (Heg2Fsbaf3		14 e	107 0.8131	
AAaa6Akf2i4f31FaJAagff4fHae44aib3f)2f22bfl6b6k2(keA312bH)4Fgcla(s1ffbU)3e12bskik		15 A	105 0.7979	
gbk2a3f4cb3is3)1aAffb4(ssgFceHisFi6A3H4fs1gki(a3kFab43fb6H3As2)	下数编集	16 s	9T 0.7371	
(dHe (HaoHadi a3Faqqaad) oHe [6]	1.000000	17 F	87 0.6611	
The strength of the strength o		10 k	86 0.6535	
LANGLINF ASLELISF WAGIESS FOR USIFIES ES LUGITIASTISCZI BESKAIZC (M. 1851 WKCOKZI 388		19 c	73 0.5548	
cAbel (32kef2) Asaggbaff) ilgefb (44761b3) 21AbcA)		a second	55 0.38	
{bgi4F2aaHAgcbakcA}cHgi2gFg6bA43]21sisilA[gkHHaHf6kasfgff]1[sabfg[sifk4c1f]ffkFH		21 46 9.00011.1107 11	48 0, 3548	
In the state of the brack for the state of t		05.60	JE 0.343	12

可以看到除了固定格式flag{}之外按顺序是6432bgHieAsFck。只能我自己拼起来可读内容了。(另外这个我后来问了一下同学, 只有我自己的词频分析是这样的,他们的词频分析出来就已经是正确的flag了)

一眼望去有bAse6432,结合图片还有kFc,最后剩下一个Hi。

所以是flag{Hibase6432kFc}

剩下我自己没有做出来的WP可以参考dalao的内容

https://mp.weixin.qq.com/s/FG5AM0PGftQuQzCl5-Es-w