




CTF密码学Crypto1

原创

宁嘉  于 2020-04-10 16:16:13 发布  1103  收藏 5

分类专栏: [BUUCTF Crypto](#) 文章标签: [信息安全](#) [密码学](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 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/MikeCoke/article/details/105392195>

版权



[BUUCTF Crypto](#) 专栏收录该内容

34 篇文章 2 订阅

订阅专栏

CTF密码学Crypto

- 1.Morse code(摩尔斯密码)
2. 栅栏密码
3. 恺撒密码
- 4.Ook!编码
- 5.Brainfuck编码
- 6.Base64编码

CTF解密工具:

1.<http://ctf.ssleye.com>

2.<https://tool.bugku.com>

1.Morse code(摩尔斯密码)

特点:

用两种“符号”来表示字符: 点 (.) 和划 (-)

Challenge

7951 Solves



滴答~滴

20

-----答案格式KEY{xxxxxxxxx}

Flag

Submit

<https://blog.csdn.net/MikeCoke>

easy_crypto

30

```
0010 0100 01 110 1111011 11 11111 010 000 0 001101 1010 111
100 0 001101 01111 000 001101 00 10 1 0 010 0 000 1 01111 10
11110 101011 1111101
```

Flag

Submit

<https://blog.csdn.net/MikeCoke>

通过写python脚本来解决:

```

ASCLL = input("Enter the string:")
L = ASCLL.split()
asc11 ={'01': 'A','1000': 'B','1010': 'C','100': 'D','0': 'E','0010': 'F','110': 'G','0000': 'H','00': 'I','0111': 'J','101': 'K','0100': 'L','11': 'M','10': 'N','111': 'O','0110': 'P','1101': 'Q','010': 'R','000': 'S','1': 'T','001': 'U','0001': 'V','011': 'W','1001': 'X','1011': 'Y','1100': 'Z','01111': '1','00111': '2','00011': '3','00001': '4','00000': '5','10000': '6','11000': '7','11100': '8','11110': '9','11111': '0','001100': '?','10010': '/','101101': '(',')','100001': '-','010101': '.', '110011': ',', '011010': '@','111000': ':','101010': ';','10001': '=', '011110': "'", '101011': '!','001101': '_','010010': '"','10110': '(',')','111101': '{','1111101': '}'
}
for item in L:
    print(asc11[item],end='')

```

摩尔斯电码与ASCLL关系:

对应电码	字符				
01	A	1100	Z		
1000	B	01111	1		
1010	C	00111	2		
100	D	00011	3		
0	E	00001	4		
0010	F	00000	5		
110	G	10000	6		
0000	H	11000	7		
00	I	11100	8		
0111	J	11110	9		
101	K	11111	0		
0100	L	001100	?		
11	M	10010	/		
10	N	100001	-		
111	O	010101	.		
0110	P	110011	,		
1101	Q	011010	@		
010	R	111000	:		
000	S	101010	;		
1	T	10001	=		
001	U	011110	'		
0001	V	101011	!		
011	W	001101	_	101101)
1001	X	010010	"	0001001	\$
1011	Y	10110	(01000	&

2. 栅栏密码

所谓栅栏密码,就是把要加密的明文分成N个一组,然后把每组的第1个字连起来,形成一段无规律的话。不过栅栏密码本身有一个潜规则,就是组成栅栏的字母一般不会太多。(一般不超过30个,也就是一、两句话)

加密原理：百度百科有详细介绍。

Challenge7574 Solves×

聪明的小羊

20

一只小羊翻过了2个栅栏KYsd3js2E{a2jda}

3. 恺撒密码

恺撒密码的替换方法是通过排列明文和密文字母表，密文字母表示通过将明文字母表向左或向右移动一个固定数目的位置。例如，当偏移量是左移3的时候（解密时的密钥就是3）：
明文字母表：ABCDEFGHIJKLMNOPQRSTUVWXYZ；
密文字母表：DEFGHIJKLMNOPQRSTUVWXYZABC。

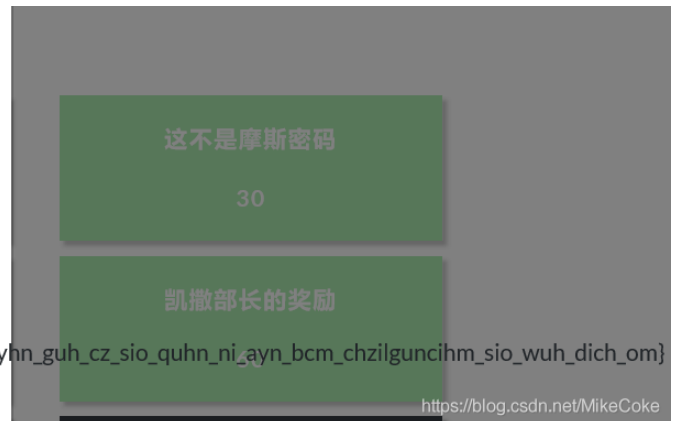
凯撒部长的奖励

60

就在8月，超师傅出色地完成了上级的特遣任务，凯撒部长准备给超师傅一份特殊的奖励，兴高采烈的超师傅却只收到一长串莫名的密文，超师傅看到英语字串便满脸黑线，帮他拿到这份价值不菲的奖励吧。密文：MSW

{byly_Cm_slol_lYqUlx_yhdls_Cn_Wuymul_il_wuff_bcg_pCwnll_cm_u_Yrwyffyh_n_guh_cz_sio_quhn_ni_ayn_bcm_chzilguncihm_sio_wuh_dich_om}

题目来源：第七季极客大挑战



4.Ook!编码

ok
30

Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook. Ook.
Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook.
Ook. Ook? Ook. Ook? Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook! Ook.
Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook?
Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook! Ook!
Ook! Ook!
Ook! Ook! Ook? Ook. Ook? Ook! Ook. Ook? Ook! Ook! Ook! Ook!
Ook! Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook.
Ook? Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook! Ook! Ook.

直接用工具解密就行。

Ook!的变形题目：

Challenge 3351 Solves ×

!?
80

..... !?! ?..... .. ??! ?... !...
..... !?..... !?! ?!!!! !!?? !?! !!!..... !?
..... !? !!?. .. ?? !?..... !?..... !?! ?!!!!
!!!! !?! ?!?. .. ! ?!?. .. ? ?!?. .. !?.....
!?! ?!!!! !!?? !?! !!!!! !!!!! .. ! ?..... !? !!?. ..
??! ?..! ?..... !?! ?!!!! !!!!! ?!?. !!!!! !!!!! !!?. ..
..!?! !?..... ? ?!?. .. ! !!!!! !!!!! !!!!! !!?. .. !?!
?... .. ? ?!?. .. !!!!! !!!!! !?..... !?! !?..... ??!
?... .. !?.

Flag

Submit

<https://blog.csdn.net/MikeCoke>

学会观察题目特点，快速突破。

5. Brainfuck 编码

特点：Base64字符表由a-z,A-Z,0-9,/,+构成,编码后末尾可能有=号

简单加密

60

e6Z9i~]8R~U~QHE{RnY{QXg~QnQ{^XVlRXlp^XI5Q6Q6SKY8jUAA

<https://blog.csdn.net/MikeCoke>

这个题就是恺撒密码与base64编码的混合加密，由字符串末尾的AA可以考虑是否为base64中的==。由Ascii知'A'与'='相差4个位，写一个python脚本将所有字符都偏移4位：

```
a = 'e6Z9i~]8R~U~QHE{RnY{QXg~QnQ{^XVlRXlp^XI5Q6Q6SKY8jUAA'
i = 0
flag = ''
while i < len(a):
    num = ord(a[i])-4
    flag += chr(num)
    i +=1
print(flag)
```

得到如下结果：

a2V5ezY4NzQzMDAwNjUwMTczMjMwZTRhNThlZTE1M2M2OGU4fQ==

用base64解密工具进行解密：

```
a2V5ezY4NzQzMdAwNjUwMTczMjMwZTRhNThlZTE1M2M2OGU4fQ==
```

编码

base64

字符集

utf8(unicode编码)

编 码

解 码

```
key{68743000650173230e4a58ee153c68e8}
```

<https://blog.csdn.net/MikeCoke>

得到Flag为：**key{68743000650173230e4a58ee153c68e8}**