

# Bugku加密writeup

原创

[TedLau](#) 于 2019-12-22 00:04:30 发布 290 收藏 1

分类专栏: [Bugku](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY](#) 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/qq\\_30204577/article/details/103649900](https://blog.csdn.net/qq_30204577/article/details/103649900)

版权



[Bugku 专栏收录该内容](#)

2 篇文章 0 订阅

订阅专栏

Bugku加密writeup

滴答~滴

这个看形式是摩斯密码, 我们可以在<https://www.mathsking.net/morse.htm> 进行摩斯密码转换, 可以得到flag: BKCTFMISC, 记住还有格式要求: KEY{}  
聪明的小羊

一只小羊翻过了2个栅栏

**KYsd3js2E{a2jda}**

题目给了明显的提示, 栅栏, 也就是栅栏密码, 所以我们进行二栏的栅栏解密可得: KEY{sad23jjdsa2}

ok

由经验可知, 下面的这种代码是Ook加密, 所以我们在网站<https://tool.bugku.com/brainfuck/?wafcloud=3%20brainfuck>, 进行Ook解密, 将文本粘贴到文本框中, 然后点击Ook! to Text按钮, 即可获得密文: flag{ok-ctf-1234-admin}



这是一种名叫brain fuck的加密方式，这个依然也可以在上面（Ook!解密)的解密网站进行解密，这次点击Brainfuck to Text，即可得到我们想要的flag: flag{ok-c2tf-3389-admin}

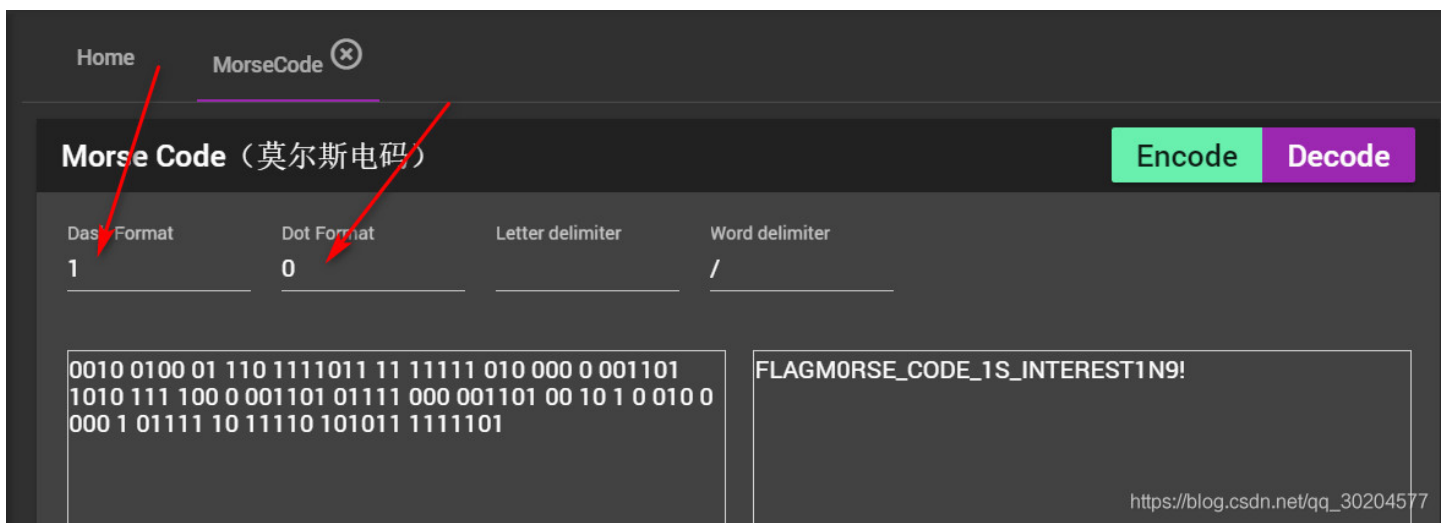
## easy\_crypto

```
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
```

这个加密，我一开始以为是ascii加密，然后进行解密后发现了{}，结果多次发现并不是这样的，于是上网看了其他大佬的blog，了解到这也是一种morse加密，只不过是把- 换成了10，我们只需要将其转换成- 然后进行普通的morse解密即可，在这里我使用的是一个现成的工具套件，因为我目前还没有学会写python脚本，打算在今年寒假学习（最近复习简直是要吐了。

解密过后如下图所示，这里我将原来的- 换成了 对应的1 0，然后便出现了FLAG字样，然而提交是错误的，将其全部改为小写字母后提交，成功了。

```
flag{m0rse_code_1s_interest1n9!}
```



## 简单加密

```
e6Z9i}8RUQHE{RnY{QXg QnQ{XMRXpXl5Q6Q6SKY8jUAA
```

不知所措。。。。

百度参考大佬的博客，了解到，这是base64加凯撒加密，最后的AA跟==很像，A: 65 =: 61，可以猜想是移了四位，然后通过脚本来解得flag，我依然不会，所有纯手工解flag，先将各位字母对应的ascii移动四位，然后解密base64，。。果然人不如机，中间错了一位，导致我重新去查看密文。。。下图是我逐个解密之前的十进制。

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

Bin  
1100101 110110 1011010 111001 1101001 1111110 1011101 111000 1010010 1111110 1010101 1111110 1010001  
1001000 1000101 1111011 1010010 1101110 1011001 1111011 1010001 1011000 1100111 1111110 1010001

Oct  
145 66 132 71 151 176 135 70 122 176 125 176 121 110 105 173 122 156 131 173 121 130 147 176 121 156 121 173 136  
130 126 154 122 130 154 160 136 130 111 65 121 66 121 66 123 113 131 70 152 125 101 101

Dec  
101 54 90 57 105 126 93 56 82 126 85 126 81 72 69 123 82 110 89 123 81 88 103 126 81 110 81 123 94 88 86 108 82 88  
108 112 94 88 73 53 81 54 81 54 83 75 89 56 106 85 65 65

Hex  
65 36 5a 39 69 7e 5d 38 52 7e 55 7e 51 48 45 7b 52 6e 59 7b 51 58 67 7e 51 6e 51 7b 5e 58 56 6c 52 58 6c 70 5e 58 49 35  
51 36 51 36 53 4b 59 38 6a 55 41 41

[https://blog.csdn.net/qq\\_30204577](https://blog.csdn.net/qq_30204577)

下图是我手动转换完之后的图示：

Base64 Encoding Encode Decode

Pattern  
Base64

a2V5ezY4NzQzMDAwNjUwMTczMjMwZTRhNThlZTE1M2M2OGU4fQ==

key{68743000650173230e4a58ee153c68e8}

[https://blog.csdn.net/qq\\_30204577](https://blog.csdn.net/qq_30204577)

下图是我手动转换完之后的图示：

嗯，，，python很重要。

未完待续~~~~

2019年12月22日00点05分