

CTF中基本的Xor解密操作

原创

[valecalida](#) 于 2019-10-24 10:36:18 发布 4504 收藏 6

分类专栏: [cmd](#) [CTF](#) [python](#) 文章标签: [Python](#) [Xor](#) [CTF](#)

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/valecalida/article/details/102717762>

版权



[cmd](#) 同时被 3 个专栏收录

4 篇文章 0 订阅

订阅专栏



[CTF](#)

21 篇文章 0 订阅

订阅专栏



[python](#)

39 篇文章 1 订阅

订阅专栏

先解密Base64, 然后进行Xor的检测及解密

```

#!/usr/bin/python3
# -*- coding: utf-8 -*-
# --author: valecalida--
# 异或运算仅允许数字之间的运算，不允许其他类型之间的运算

from base64 import b64decode as b64d
message = input("请输入您想要进行操作的字符串 >>>")
if message[0:2] == "b\\":
    message = message[2:-1]
#print(message)
flags = input("请输入解码的样式(例: flag、ctfhub) >>>")

def b64_detect(msg):
    try:
        cipher_text = b64d(msg)
    except BaseException as e:
        print("您输入的值好像不能使用Base64解密,请再尝试别的方法")
    else:
        res = []
        for i in range(len(flags)):
            res.append(cipher_text[i] ^ ord(flags[i]))
    finally:
        return res, cipher_text

def decode_xor():
    result = ''
    res, cipher_text = b64_detect(message)
    if res[0] - res[1] == 0:
        print("这是一个值不变的Xor运算")
        for i in range(len(cipher_text)):
            result += chr(res[0] ^ cipher_text[i])
        return result
    elif res[0] - res[1] == 1:
        print("这是一个值递减的Xor运算")
        for i in range(len(cipher_text)):
            result += chr((res[0] - i) ^ cipher_text[i])
        return result
    elif res[0] - res[1] == -1:
        print("这是一个值递增的Xor运算")
        for i in range(len(cipher_text)):
            result += chr((res[0] + i) ^ cipher_text[i])
        return result
    else:
        print("这好像不是Xor运算,再试试别的吧")
        return result

print("\t程序返回的结果是 >>", decode_xor())

```

运行结果如下：

请输入您想要进行操作的字符串 >>>b'cXR4fWB0eHJzTlw='

cXR4fWB0eHJzTlw=

请输入解码的样式(例: flag、ctfhub) >>>flag

这是一个值递增的Xor运算

程序返回的结果是 >> flag{hello}

进程已结束, 退出代码 0

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

这是一个接触到的OTP类型的题目,但是我感觉好像没考到这个知识点(黑人问号脸???),为了避免侵权,这里打上马赛克(侵权请联系我)

```
mtpxor.txt - 记事本
文件(E) 编辑(E) 格式(O) 查看(V) 帮助(H)
Hint:The [REDACTED] ht
=====
24161a1 [REDACTED] 0f1818001e06120c03170e
380e045 [REDACTED] 040154121a0e120100071c
2511531 [REDACTED] 90916030a000302581c1d15
3c0818 [REDACTED] 0c1c1a011b01460c07175d
24161 [REDACTED] 0f1818001e06120c03170e
380e045 [REDACTED] 40154121a0e120100071c
2709161 [REDACTED] 0091d0b151c131606011a
2709161 [REDACTED] 91a1a0201070f160a0108
2409161 [REDACTED] 60a1b1000030f0c1b1e18
24161a1 [REDACTED] 18180013030a0c071713
2409161 [REDACTED] 161809171d0f161b1a18
2409121 [REDACTED] 010d0a071d1211010b0e
330e061 [REDACTED] 654121a060510181304
https://blog.csdn.net/valecalida
```

由于题目已经给出了hint,所以这里直接用就行了

```
#!/usr/bin/python2
# -*- coding: utf-8 -*-
import binascii
c1 = '24161a1d1*****20c03170e'
c2 = '380e*****120100071c'
c3 = '2511*****000302581c1d15'
c4 = '1*****1a**b01460c07175d'
c5 = '24161a1*****06120c03170e'
c6 = '380e*****0e120100071c'
c7 = '270*****606011a'
c8 = '27091*****f60a0108'
c9 = '24090*****0030f0c1b1e18' # 这是密码
c10 = '24161*****13030a0c071713'
c11 = '2409*****161b1a18'
c12 = '24091*****1d1211010b0e'
c13 = '330e06*****60510181304'
ciphers = [c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11,c12,c13]
cipher_text = "Th*****ht"

def sxor(s1,s2):
    return ''.join(chr(ord(a) ^ ord(b)) for a,b in zip(s1, s2))

for cipher in ciphers:
    k = sxor(cipher.decode('hex'),cipher_text)
    print(binascii.a2b_hex(k.encode('hex')))
```

注意这里用的是python2

运行结果

```
p~ selbcdKxawulj {xw {k
lfa=G px}Fsx;ghb {utky
qy6|loqr3V~ol rok,hqp
h~}x.a'szC{eutim/xs {8
p~ selbcdKxawulj {xw {k
lfa=G px}Fsx;ghb {utky
sass.torqNwpr gpzbrm
sass.hby|V~cuwskfb~mm
password
p~ selbcdKxawuaocxs {v
passzhbcaC ow|eqfbov}
pawses~xfDyxb uq {eugk
gfcqj or3Qso;ghjldl a
```

这里直接得到了密码,所以我也不知道它到底有没有考到这个知识点,但是个人感觉没有,给大家看着玩吧



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)