CTF题目中XOR加密解法脚本

原创

神林、 于 2019-04-10 21:59:53 发布 8111 收藏 4

分类专栏： CTF

本文链接：https://blog.csdn.net/qq_41079177/article/details/89196428

CTF 专栏收录该内容

24 篇文章 2 订阅

订阅专栏

0x00:xor加密原理

简单异或密码（英语：simple XOR cipher）是密码学中一种简单的加密算法，它按照如下原则进行运算：

A {\displaystyle \oplus } \oplus 0 = A

A {\displaystyle \oplus } \oplus A = 0

(A {\displaystyle \oplus } \oplus B) {\displaystyle \oplus } \oplus C = A {\displaystyle \oplus } \oplus (B {\displaystyle \oplus } \oplus C)

(B {\displaystyle \oplus } \oplus A) {\displaystyle \oplus } \oplus A = B {\displaystyle \oplus } \oplus 0 = B

其中 {\displaystyle \oplus } \oplus 为逻辑异或（XOR）运算的符号。按这种逻辑，文本串行的每个字符可以通过与给定的密钥进行按位异或运算来加密。如果要解密，只需要将加密后的结果与密钥再次进行按位异或运算即可。

例如，字符串“Wiki”（8位ASCII：01010111 01101001 01101011 01101001） 可以按如下的方式用密钥11110011进行加密：

01010111 01101001 01101011 01101001

{\displaystyle \oplus } \oplus 11110011 11110011 11110011 11110011

= 10100100 10011010 10011000 10011010

此种加密方法类似对称加密，故解密的方式如下：

10100100 10011010 10011000 10011010

{\displaystyle \oplus } \oplus 11110011 11110011 11110011 11110011

= 01010111 01101001 01101011 01101001

异或运算符常作为更为复杂的加密算法的组成部分。对于其本身来说，如果使用不断重复的密钥，利用频率分析就可以破解这种简单的异或密码。如果消息的内容被猜出或知道，密钥就会泄露。异或密码值得使用的原因主要是其易于实现，而且计算成本小。简单重复异或加密有时用于不需要特别安全的情况下来隐藏信息。

如果密钥是随机的（不重复），而且与消息长度相同，异或密码就会更为安全。当密钥流由伪随机数发生器生成时，结果就是流密码。若密钥是真正随机的，结果就是一次性密码本，这种密码在理论上是不可破解的。

在这些密码的任何部分中，密钥运算符在已知明文攻击下都是脆弱的，这是因为明文 {\displaystyle \oplus } \oplus 密文 = 密钥。

0x01:xor,python解密脚本

已知明文和密文：

```
m1 = open('密文.txt','rb').read()
m2 = open('明文.txt','rb').read()
mes = "".join(map(lambda item:chr(item[0]^item[1]),list(zip(m1,m2))))
print(mes)
```

1.明文密文都用二进制的方式读取
2.对两个数据进行按位异或
3.将异或的二进制转换成字符串
4输出字符串
map：

```
# 提供了两个列表，对相同位置的列表数据进行相加
>>> map(lambda x, y: x + y, [1, 3, 5, 7, 9], [2, 4, 6, 8, 10])
[3, 7, 11, 15, 19]
```

lambad:作为匿名函数存在，对其进行异或并且转换为字符串形式
zip：

```
>>>a = [1,2,3]
>>> b = [4,5,6]
>>> c = [4,5,6,7,8]
>>> zipped = zip(a,b)     # 返回一个对象
>>> zipped
```

list(zip()):

```
>>> list(zipped)  # list() 转换为列表#python3中转化成列表使用list函数，python2默认返回列表
[(1, 4), (2, 5), (3, 6)]
```

0x02:python加密脚本

根据异或原理A xor B = C;
C xor B = A；

```
m1 = open('明文.txt','rb').read()
m2 = open('真实明文.txt','rb').read()
mes = "".join(map(lambda item:chr(item[0]^item[1]),list(zip(m1,m2))))
m3 = open('密文.txt','w')
m3.write(mes)
m3.close()
```