攻防世界 Crypto高手进阶区 3分题 streamgame1

原创

思源湖的鱼 于 2020-12-09 13:39:05 发布 284 收藏

分类专栏： ctf 文章标签： ctf 攻防世界 crypto

本文链接：https://blog.csdn.net/weixin_44604541/article/details/110920752

前言

继续ctf的旅程

攻防世界Crypto高手进阶区的3分题

本篇是streamgame1的writeup

发现攻防世界的题目分数是动态的

就仅以做题时的分数为准了

解题过程

得到一个key文件

和一段python

```
from flag import flag
assert flag.startswith("flag{")
# 作用：判断字符串是否以指定字符或子字符串开头flag{
assert flag.endswith("}")
# 作用：判断字符串是否以指定字符或子字符串结尾}，flag{}，6个字节
assert len(flag)==25
# flag的长度为25字节，25-6=19个字节
#3<<2可以这么算，bin(3)=0b11向左移动2位变成1100，0b1100=12(十进制)
def lfsr(R,mask):
    output = (R << 1) & 0xffffff     #将R向左移动1位，bin(0xffffff)='0b111111111111111111111111'=0xffffff的二进制补
码
    i=(R&mask)&0xffffff              #按位与运算符&：参与运算的两个值,如果两个相应位都为1,则该位的结果为1,否则为0
    lastbit=0
    while i!=0:
        lastbit^=(i&1)     #按位异或运算符：当两对应的二进位相异时，结果为1
        i=i>>1
    output^=lastbit
    return (output,lastbit)



R=int(flag[5:-1],2)
mask    =    0b10100110001000110100

f=open("key","ab")    #以二进制追加模式打开
for i in range(12):
    tmp=0
    for j in range(8):
        (R,out)=lfsr(R,mask)
        tmp=(tmp << 1)^out    #按位异或运算符：当两对应的二进位相异时，结果为1
    f.write(chr(tmp))  #chr() 用一个范围在 range（256）内的（就是0～255）整数作参数，返回一个对应的字符。
f.close()
```

也就是说

- flag是19位的二进制
- 给了个加密函数
- 对mask和R作用加密函数并生成新的R同时得到1bit数据，然后每8bit数据转化成一个对应的ascii再写入key文件中

key扔进winhex

```
00000000  59 38 F7 42 C1 0D B2 C7  ED E0 24 3A              Y8÷BÁ ²Çíà$:
```

那就爆破就完事了

```python
def check(list1, list2):
    for i in range(12):
        if list1[i] != list2[i]:
            return False
    return True


def lfsr(R ,mask):
    output = (R << 1) & 0xffffff    #将R向左移动1位，bin(0xffffff)='0b111111111111111111111111'=0xffffff的二进制补码
    i=(R&mask)&0xffffff             #按位与运算符&：参与运算的两个值,如果两个相应位都为1,则该位的结果为1,否则为0
    lastbit=0
    while i!=0:
        lastbit^=(i&1)    #按位异或运算符：当两对应的二进位相异时，结果为1
        i=i>>1
    output^=lastbit
    return (output,lastbit)


if __name__ == '__main__':
    f = open('key','rb')
    content = f.read()
    s_list = []
    for c in content:
        s_list.append(c)

    print(s_list)

    mask = 0b10100011000100011100

    for i in range(1 << 19):
        print(i)
        tmp_list = []
        R = i
        for j in range(12):
            tmp = 0
            for k in range(8):
                (R, out) = lfsr(R, mask)
                tmp = (tmp << 1) ^ out  # 按位异或运算符：当两对应的二进位相异时，结果为1
            tmp_list.append(tmp)

        if (check(s_list, tmp_list)):
            print(bin(i))
```

得到flag：`flag{11101011000011010111}`

结语

简单爆破