2018 NSEC crypto leak writeup

原创

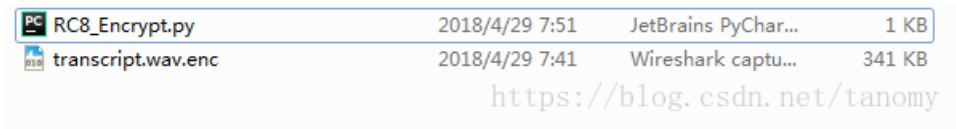
tanomy 于 2018-08-24 19:56:42 发布 651 收藏

文章标签： CTF Crypto 安全技术 密码学 流密码

本文链接：https://blog.csdn.net/tanomy/article/details/81952339

1.题目分析

题目可以在此处下载。下载并解压后会得到一个加密python脚本和一个加密后的wav文件。



题目描述

> A recent transcript suspected to contain incriminating conversations between the officials in charge of the Mars colonization plan has leaked but is encrypted with the latest top-secret encryption algorithm: RC8! You've recovered the source code to their new encryption algorithm, but the key and seed value are missing. Find a weakness in the scheme and recover the transcript.

大概意思就是说希望我们在没有key和seed的情况下去解密文件。为此，我们先来看看RC8_Encrypt.py中的代码了解其是如何进行加密的。

```python
#!/usr/bin/env python3
import sys

def rc8(state, key, n):
    '''
    Top Secret RC8 Stream Cipher
    '''
    while (n > 0):
        yield state & 0xff
        for _ in range(8):
            c, s = key, state
            b = 0
            while c:
                b ^= c & 1 * s & 1
                c >>= 1 ; s >>= 1
            state = state >> 1 | b << 63
        n -= 1

def main():
    seed, key = ?, ? # Missing

    with open(sys.argv[1], 'rb') as fin:
        data = bytearray(fin.read())

    for i,x in enumerate(rc8(seed, key, len(data))):
        data[i] ^= x

    with open(sys.argv[1] + '.enc', 'wb') as fout:
        fout.write(data)

if __name__ == "__main__":
    main()
```

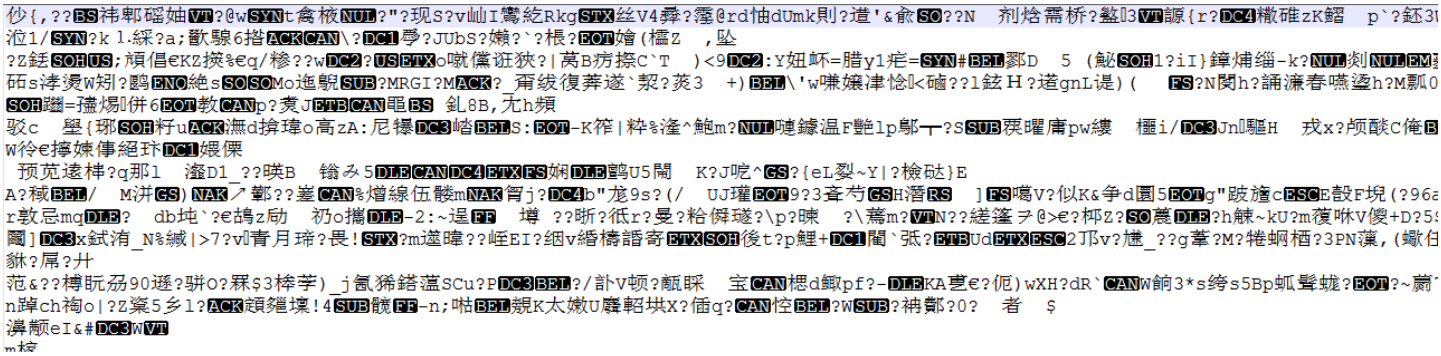
我们可以很容易看出这是一种流加密，密钥流生成函数是rc8()，也很容易看出密钥流每次取得是state的最低八位，下面主要分析一下state是如何变化的，变化的代码如下

```python
for _ in range(8):
            c, s = key, state
            b = 0
            while c:
                b ^= c & 1 * s & 1
                c >>= 1 ; s >>= 1
            state = state >> 1 | b << 63
```

设

$$\times$$

大致了解了加密算法后，我们再来看看加密后的文件能给我们什么信息。打开一看，全是乱码，没有什么参考价值，当然这也是我们意料之中的事。



我们看到这道题目的名字是crypto leak，那么到底leak（泄露）了什么呢？我们知道我们加密的文件是wav格式的，而每种格式的文件都是有其特定的文件头的，我们来看看wav文件头是什么样的。

位置	字节数	值/类型	描述
1-4	4	“RIFF”	表示文件”RIFF”文件
5-8	4	Uint32	文件长度-8
9-12	4	“WAVE”	文件类型头，表示一个”WAVE”文件
13-16	4	“fmt “	格式表示符

我们知道加密前的文件是wav格式的文件，因此文件的前16字节我们就都可以得到了，其中文件长度加密后的长度和加密前是一样的，我们可以直接得到

```
>>>import os
>>>hex(os.path.getsize("transcript.wav.enc")-8)
>>>'0x55030L'
```

注意Uint32类型的文件大小值在计算机中一般是小端保存的，因此这四个字节应该是30 50 05 00。加密前文件的前16字节是：
52 49 46 46 30 50 05 00 57 41 56 45 66 6D 74 20
同时我们也可以通过010editor打开加密后的文件查看前16字节的密文：
81 A3 7B 2C CD 36 EB A0 08 B5 74 DB AF B4 74 E6
用那么将它们异或就可以得到前16次加密的密钥流了:
D3 EA 3D 6A FD 66 EE 30 5F F4 22 9E C9 D9 00 C6

2.破解加密

我们已经得到泄露的前16次加密的密钥流了：
D3 EA 3D 6A FD 66 EE 30 5F F4 22 9E C9 D9 00 C6
我们根据代码不难分析出前8次加密的秘钥流就是seed的值，因此seed=B6F2E93D1C6F3A08。那么接下来我们应该想想如何把key求出来，我们根据前面的分析，知道state每次向右移一位，而新的最高位的计算公式是：

=

新的