

# 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](#)

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/tanomy/article/details/81952339>

版权

## 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。

 RC8_Encrypt.py	2018/4/29 7:51	JetBrains PyChar...	1 KB
 transcript.wav.enc	2018/4/29 7:41	Wireshark captu...	341 KB

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

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。

```

#!/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

```

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

`  
  x

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。

沙{, ??BS柿軒福姐VT?@wSYNt禽板NUl?"?现S?v山I鷄絃RkgSTX丝V4舞?霍@rd恤dUmk則?道'&俞SO??N 剂熒需桥?整l3VT諺(r?DC4撒硅zK鉛 p`?鉛31泣@1/SYN?k 1.深?a;歡驥6搭ACKCAN ?DC1夢?JUBs?嬪? ?根?EDN增(極Z , 跌  
?2銓SORUS;頗信€KZ換%Eq'穆??wDC2?USETXo吼黨斑狹?|萬B房擦C`T )<9DC2:Y姍虾=腊y1疮=SYN#BEL鄧D 5 (鯀SOH!iI}鐘捕繩-k?NUL剝NULEM  
砧s淳透W知?鷗ENO絕sSOSONo迤貌SUB?MRG1?MACK? 遷級復奔遂、梨?葵3 +) BEL\ 'w嫌娘津恰?確?1鉢H ?道gnL提) ( ES?N閔h?誦濂春嚙塗h?M瓢0  
SOH趨=撫煩2併6EOT教CAN?黃JETBCAN電BS 釣B, 先h頻  
駁c 塵{那SOH籽uACK?無d揜瑋o高ZA:尼撫DC3!咱BELS:EOT-K作|粹%逢^鮑m?NUL噴錄溫F艷1p鄧-+?S SUB表曜庸pw縷 檻i/DC3JnU驅H 戎x?頏酸C俺B  
W伶e摶棘僅組环DC1媛櫻  
預覓遠極?q那1 隘D1\_??喚B 鑰み5DLE|CANDC4ETXFS炳DLE鷄U5閩 K?J吃^GS?(eL裂~Y|?檢碰)E  
A?穢BEL/ M讲GS) NAK / 鄭??臺CAN%增線伍餸mNAK晉j?DC4b"龍9s?(/ UJ3舊EOT923查苟GSH潛RS ]ES噶v?似k&爭d圓5EOTg"跋旗cESCE設F境(?96a  
r敦忌mqDLE? db坤`e鵠z勵 初o攜DLE-2:~逞FE 增 ??晰?低r?曼?給僻遜?p?暎 ?\蕪m?WIN??綾篷矛@>e?桺Z?SO農DLE?h鍊~kU?m楓琳V傻+D?5  
國]DC3x試消\_N%誠|>7?v[青月瑞?畏!STX?m達曉??蛭EI?網v緒橋諳寄ETXSOH後t?p鯀+DC1閩、張?ETBUdETXESC2Jbv?達\_??g葦?M?捲婀栖?3PN漢,(蠍伯  
獅?屏?升  
范??博耽弱90遜?駢o?釋\$3棒芋)\_j氤豨錯蘊SCu?PDC3BEL/?計V頓?瓶睬 宝CAN標d鐵pf?-DLEKA臺e?厄)wxH?dR `CANW餉3\*s綺s5Bp虹聳蝶?EOT?~蔚  
n躁ch掏o|?2算5乡1?ACK趙鑑壞!4SUB競FB-n;咷BEL競K太嫩u驕軒X?值q?CAN腔BEL?wSUB?衲鄭?0? 者 \$  
溝顛eI & #DC3nVT  
m大文

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

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

