RE-实验吧/有一个程序加密得到如下密文

原创

Alikas 于 2019-04-14 12:28:20 发布 216 收藏

分类专栏： 逆向 文章标签： 实验吧 writeup RE

本文链接：https://blog.csdn.net/q83182034/article/details/89294948

Alikas-0x06

题目：*实验吧 有一个程序加密得到如下密文*

看到pyc文件肯定是反编译的啦，这里我用的是`uncompyle2`。（再次怒夸子系统）

安装过程如下：

```
git clone https://github.com/wibiti/uncompyle2

cd uncompyle2

python setup.py install
```

然后进行反编译

```
uncompyle2 reverse300.pyc > reverse300.py
```

接下来打开reverse300.py查看代码

```
# 2019.04.13 21:35:38 DST
#Embedded file name: ./rev200.py
import sys
from hashlib import md5
import base64
from time import time
from datetime import datetime
UC_KEY = '123456789'

def authcode(string, operation = 'DECODE', key = UC_KEY, expiry = 0):
    ckey_length = 4
    if key == '':
        key = md5(UC_KEY.encode('utf-8')).hexdigest()
    else:
        key = md5(key.encode('utf-8')).hexdigest()
    keya = md5(key[0:16].encode('utf-8')).hexdigest()
    keyb = md5(key[16:32].encode('utf-8')).hexdigest()
    if ckey_length == 0:
        keyc = ''
    elif operation == 'DECODE':
        keyc = string[0:ckey_length]
    elif operation == 'ENCODE':
```

```
        keyc = md5(str(datetime.now().microsecond).encode('utf-8')).hexdigest()[-ckey_length:]
    else:
        return
    cryptkey = keya + md5((keya + keyc).encode('utf-8')).hexdigest()
    key_length = len(cryptkey)
    if operation == 'DECODE':
        string = base64.b64decode(string[ckey_length:])
    elif operation == 'ENCODE':
        if expiry == 0:
            string = '0000000000' + md5((string + keyb).encode('utf-8')).hexdigest()[0:16] + string
        else:
            string = '%10d' % (expiry + int(time())) + md5((string + keyb).encode('utf-8')).hexdigest()[0:16] +
string
    else:
        return
    string_length = len(string)
    result = ''
    box = range(256)
    rndkey = [0] * 256
    for i in range(256):
        rndkey[i] = ord(cryptkey[i % key_length])

    j = 0
    for i in range(256):
        j = (j + box[i] + rndkey[i]) % 256
        tmp = box[i]
        box[i] = box[j]
        box[j] = tmp

    a = j = 0
    for i in range(string_length):
        a = (a + 1) % 256
        j = (j + box[a]) % 256
        tmp = box[a]
        box[a] = box[j]
        box[j] = tmp
        result += chr(ord(string[i]) ^ box[(box[a] + box[j]) % 256])

    if operation == 'DECODE':
        if not result[0:10].isdigit() or int(result[0:10]) == 0 or int(result[0:10]) - int(time()) > 0:
            if result[10:26] == md5(result[26:].encode('utf-8') + keyb).hexdigest()[0:16]:
                return result[26:]
            else:
                return ''
        else:
            return ''
    else:
        return keyc + base64.b64encode(result)


if __name__ == '__main__':
    if len(sys.argv) < 3:
        exit(1)
    ex = 20
    for i in range(1, len(sys.argv), 2):
        a = sys.argv[i]
        b = sys.argv[i + 1]
        if a == '-t':
            ex = int(b)
        elif a
```

```
        elif a == '-e':
            encoded = authcode(b, 'ENCODE', expiry=ex)
            print encoded
        elif a == '-d':
            decoded = authcode(b, 'DECODE', expiry=ex)
            print decoded
+++ okay decompyling reverse300.pyc
# decompiled 1 files: 1 okay, 0 failed, 0 verify failed
# 2019.04.13 21:35:38 DST
```

解法一：

查看主函数，因为涉及解密，那直接去看decode。

看到`return ''`直接进行修改，把后两个`return ''`改为`return result[26:],`保存。

核心代码如下：

```
if operation == 'DECODE':
        if not result[0:10].isdigit() or int(result[0:10]) == 0 or int(result[0:10]) - int(time()) > 0:
            if result[10:26] == md5(result[26:].encode('utf-8') + keyb).hexdigest()[0:16]:
                return result[26:]
            else:
                return ''   #修改这里
        else:
            return ''      #和这里
    else:
        return keyc + base64.b64encode(result)
```

此时执行

```
python reverse300.py -d 0be6770IigHXZpz9hQYR1fpl15R0z9MUalmYEPhJeEN/sRklL6wQw5yQ7SAyT6tKGJNY0AxnyzS/L7zWQII=
```

然后就掉坑里了

```
python reverse300.py -d 0be6770
IigHXZpz9hQYR1fpl15R0z9MUalmYEPhJeEN/sRklL6wQw5yQ7SAyT6tKGJNY0AxnyzS/L7zWQII=
  File "reverse300.py", line 87
    +++ okay decompyling reverse300.pyc
                        ^
SyntaxError: invalid syntax
```

改一下，完美运行！

```
python reverse300.py -d 0be
6770IigHXZpz9hQYR1fpl15R0z9MUalmYEPhJeEN/sRklL6wQw5yQ7SAyT6tKGJNY0AxnyzS/L7zWQII=
DUTCTF{2u0_chu_14i_d3_5hi_h3n74i}
```

解法二：

分析加密算法自己写出解密算法：（这是大佬教的！tql@-@!）

```
from hashlib import md5
from Crypto.Cipher import ARC4
UC_KEY = '123456789'
key = md5(UC_KEY.encode('utf-8')).hexdigest()
cipher = '0be6770IigHXZpz9hQYR1fpl15R0z9MUalmYEPhJeEN/sRklL6wQw5yQ7SAyT6tKGJNY0AxnyzS/L7zWQII='
keyc = cipher[0:4]
cipher = cipher[4:]
cipher = cipher.decode('base64')
keya = md5(key[0:16].encode('utf-8')).hexdigest()
keyb = md5(key[16:32].encode('utf-8')).hexdigest()
cryptkey = keya + md5((keya + keyc).encode('utf-8')).hexdigest()
rc4 = ARC4.new(cryptkey)
print(rc4.decrypt(cipher))
```

总结：

1.这道题不难，但是我从未遇见过反编译pyc文件，一开始有点懵，后来查到可以有`uncompyle2`这种神奇工具！Nice！学到了！

2.然后去问大佬，大佬提供了第二种解法，直接分析加密过程，ARC4和md5，算是初步开始接触现代密码了吧！

（dalao说这种加密要一眼就能看出来啊，任重而道远呀...）