最好的语言

发表于 2021-01-04 分类于 Challenge， 2018， 网鼎杯， 第三场， Reverse
Challenge | 2018 | 网鼎杯 | 第三场 | Reverse | 最好的语言

点击此处获得更好的阅读体验

来源

来自Prowes5原创投稿

题目考点

- *pyc恢复*
- *md5、base64加密算法*

解题思路

下载下来题目，发现是一个类似于xml格式的文本文件。看到头部有magic字段。

```
magic 03f30d0a
moddate 6d4a695b (Tue Aug  7 15:29:49 2018)
<code>
   <argcount> 0 </argcount>
   <nlocals> 0</nlocals>
   <stacksize> 5</stacksize>
   <flags> 0040</flags>
   <code>
   ......
```

看出这个应该是pyc文件生成的，在网上一番查找。找到了这个项目https://github.com/hinus/railgun/blob/master/src/main/python/rgparser/show.py

先编译一个简单的pyc测试一下。

```
# filename: 1.py
import base64
a = 1
b = 2
```

编译并解析pyc

```
python -m py_compile 1.py
python show.py 1.pyc
```

输出如下效果

```
magic 03f30d0a
moddate b272325f
<code>
   <argcount> 0 </argcount>
   <nlocals> 0</nlocals>
   <stacksize> 2</stacksize>
   <flags> 0040</flags>
   <code> 6400006401006c00005a00006402005a01006403005a020064010053</code>
   <dis>
  1           0 LOAD_CONST               0 (-1)
              3 LOAD_CONST               1 (None)
              6 IMPORT_NAME              0 (base64)
              9 STORE_NAME               0 (base64)
  3          12 LOAD_CONST               2 (1)
             15 STORE_NAME               1 (a)
  4          18 LOAD_CONST               3 (2)
             21 STORE_NAME               2 (b)
             24 LOAD_CONST               1 (None)
             27 RETURN_VALUE
   </dis>
   <names> ('base64', 'a', 'b')</names>
   <varnames> ()</varnames>
   <freevars> ()</freevars>
   <cellvars> ()</cellvars>
   <filename> '1.py'</filename>
   <name> '<module>'</name>
   <firstlineno> 1</firstlineno>
   <consts>
      -1
      None
      1
      2
   </consts>
   <lnotab> 0c020601</lnotab>
</code>
```

可以很明显的看到，这里是有dis也就是字节码字段的，猜测出题人把这段删除了。没有了字节码，还原没有那么容易，不过由于还有其他字段，也可以进行还原，最后比较一下二进制就可以了。大致逻辑清楚之后可能会有某些细节解析文件中是不存在的，这时候就可以对比二进制数据进行尝试。

就这样不断的尝试，编译，解析，一顿操作之后，终于还原出来了大部分代码。[原文件名为re2.py](#)。

```
import base64
from hashlib import md5
import random
import string
f = 'flag{*******}'
def _(b):
    __ = ''.join(random.sample(string.digits,4))
    ___ = ''
    for i in range(len(b)):
        ___ += chr(ord(b[i])^ord(__[i % 4]))
    return ___
def ____(a):
    ___ = md5()
    ___.update(a)
    return ___.digest()
e = _(f[:12])+____(f[12:19])+_(f[19:])
print base64.b64encode(e)
e = 'U1VQU05pSHdqCEJrQu7FS7Vngk1OTQ58qqghXmt2AUdrcFBBUEU='
```

有了源代码，写一下解密代码问题也不大，可以看到_()的加密与解密之后的位数是相同的。所以就可以分成三段解密，前边一段由于flag格式为`flag{`，所以随机的key反推出来也不是问题。而中间的是一段md5，拿出来转成32位，网上解密即可。而对于后面这部分，除了最后一位是}的信息，其他都不知道，所以需要将符合条件的全部输出出来，去找看上去像flag的。

```
import base64
e = 'U1VQU05pSHdqCEJrQu7FS7Vngk1OTQ58qqghXmt2AUdrcFBBUEU='
e = base64.b64decode(e)
print 'md5: '+e[12:12+16].encode('hex')
before = e[0:12]
after = e[28:]
for i in range(0x30, 0x3a):
    for i1 in range(0x30, 0x3a):
        for i2 in range(0x30, 0x3a):
            for i3 in range(0x30, 0x3a):
                tmp = ''
                key = chr(i)+chr(i1)+chr(i2)+chr(i3)
                for a in range(len(after)):
                    tmp += chr(ord(after[a]) ^ ord(key[a % 4]))
                #if tmp[0:4] == 'flag':
                #    print tmp
                if tmp[-1] == '}':
                    print tmp
                    #_N0t_Hard}
```

这里就给出爆破后一段flag的脚本，爆破前一段的改一下就行，不爆破也行。

FLAG

```
flag{PyC_1s_613u21i_N0t_Hard}
```

- ***本文作者：*** *CTFHub*
- ***本文链接：*** *https://writeup.ctfhub.com/Challenge/2018/网鼎杯/第三场/Reverse/hF9r3aJKC7d2uxwZMa85gp.html*
- ***版权声明：*** *本博客所有文章除特别声明外，均采用 BY-NC-SA 许可协议。转载请注明出处！*

 # Challenge # 2018 # Reverse # 网鼎杯 # 第三场
re1
beijing