

# 攻防世界 Misc高手进阶区 4分题 Py-Py-Py

原创

思源湖的鱼 于 2020-12-09 13:28:47 发布 518 收藏 1

分类专栏: [ctf](#) 文章标签: [ctf](#) [攻防世界](#) [misc](#) [pyc隐写](#)

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/weixin\\_44604541/article/details/110920403](https://blog.csdn.net/weixin_44604541/article/details/110920403)

版权

## CTF

[ctf 专栏收录该内容](#)

200 篇文章 23 订阅

订阅专栏

### 前言

继续ctf的旅程

攻防世界Misc高手进阶区的4分题

本篇是Py-Py-Py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pyc

反编译

```
# uncompile6 version 3.5.0
# Python bytecode 3.6 (3379)
# Decompiled from: Python 2.7.5 (default, Aug 7 2019, 00:51:29)
# [GCC 4.8.5 20150623 (Red Hat 4.8.5-39)]
# Embedded file name: pystego.py
# Compiled at: 2017-08-01 00:44:47
# Size of source mod 2**32: 1961 bytes
import sys, os, hashlib, time, base64
flag = '9474yeUMWODKruX70FzD9oek028+EqYcZhrUjwNm92NSU+eYXOPsRPEFrNMs7J+4qautoqOrvq28pLU='

def crypto(string, op='encode', public_key='ddd', expirytime=0):
    ckey_lenth = 4
    public_key = public_key and public_key or ''
    key = hashlib.md5(public_key).hexdigest()
    keya = hashlib.md5(key[0:16]).hexdigest()
    keyb = hashlib.md5(key[16:32]).hexdigest()
    keyc = ckey_lenth and (op == 'decode' and string[0:ckey_lenth] or hashlib.md5(str(time.time())).hexdigest()[
32 - ckey_lenth:32]) or ''
```

```

cryptkey = keya + hashlib.md5(keya + keyc).hexdigest()
key_lenth = len(cryptkey)
string = op == 'decode' and base64.b64decode(string[4:]) or '000000000' + hashlib.md5(string + keyb).hexdigest()[0:16] + string
string_lenth = len(string)
result = ''
box = list(range(256))
randkey = []
for i in xrange(255):
    randkey.append(ord(cryptkey[(i % key_lenth)]))

for i in xrange(255):
    j = 0
    j = (j + box[i] + randkey[i]) % 256
    tmp = box[i]
    box[i] = box[j]
    box[j] = tmp

for i in xrange(string_lenth):
    a = j = 0
    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 op == 'decode':
    if result[0:10] == '000000000' or int(result[0:10]) - int(time.time()) > 0:
        if result[10:26] == hashlib.md5(result[26:] + keyb).hexdigest()[0:16]:
            pass
            return result[26:]
        else:
            return
    else:
        return keyc + base64.b64encode(result)

if __name__ == '__main__':
    while True:
        flag = raw_input('Please input your flag:')
        if flag == crypto(flag, 'decode'):
            print('Success')
            break
        else:
            continue

```

把flag输出

```
1 # uncompile6 version 3.5.8
2 # Python bytecode 3.6 (3379)
3 # Decompiled from: Python 2.7.5 (default, Aug 7 2019, 00:51:29)
4 # [GCC 4.8.5 20150623 (Red Hat 4.8.5-39)]
5 # Embedded file name: pystego.py
6 # Compiled at: 2017-08-01 00:44:47
7 # Size of source mod 2**32: 1961 bytes
8 import sys, os, hashlib, time, base64
9 flag = '9474yeUMODKruX7OfzD9oek028+EqYCZhrUjWm92NSU+eYXOPsRPEFrNMs7J+4qautqOrvq28PLU-'
10
11 def crypto(string, op='encode', public_key='ddd', expirytime=0):
12     ckey_lenh = 4
13     public_key = public_key and public_key or ''
14     key = hashlib.md5(public_key).hexdigest()
15     keya = hashlib.md5(key[0:16]).hexdigest()
16     keyb = hashlib.md5(key[16:32]).hexdigest()
17     keyc = ckey_lenh and (op == 'decode' and string[0:ckey_lenh] or hashlib.md5(str(time.time())).hexdigest()[32 - ckey_lenh:32]) or ''
18     cryptkey = keya + hashlib.md5(keya + keyc).hexdigest()
19     key_lenh = len(cryptkey)
20     string = op == 'decode' and base64.b64decode(string[4:]) or '0000000000' + hashlib.md5(string + keyb).hexdigest()[0:16] + string
21     string_lenh = len(string)
22     result = ''
23     box = list(range(256))
24     randkey = []
25     for i in xrange(255):
26         randkey.append(ord(cryptkey[(i % key_lenh)]))
27
28     for i in xrange(255):
29         j = 0
30         j = (j + box[i] + randkey[i]) % 256
31         tmp = box[i]
32         box[i] = box[j]
33         box[j] = tmp
34
35     for i in xrange(string_lenh):
36         a = j = 0
37         a = (a + 1) % 256
38         j = (j + box[a]) % 256
39         tmp = box[a]
40         box[a] = box[j]
41         box[j] = tmp
42         result += chr(ord(string[i]) ^ box[((box[a] + box[j]) % 256)])
43
44     if op == 'decode':
45         if result[0:10] == '0000000000' or int(result[0:10]) - int(time.time()) > 0:
46             if result[10:26] == hashlib.md5(result[26:] + keyb).hexdigest()[0:16]:
47                 pass
48             return result[26:]
49         else:
50             return
51     else:
52         return keyc + base64.b64encode(result)
53
54
55 if __name__ == '__main__':
56     print(crypto(flag, 'decode'))
```

The challenge is Steganography  
sandbox> exited with status 0

[https://blog.csdn.net/weixin\\_44604541](https://blog.csdn.net/weixin_44604541)

结果是个提示

隐写

查了查

找到

Stegosaurus

```
cy@kali:~/stegosaurus$ python3 stegosaurus.py -x 1.pyc
Extracted payload: Flag{HiD3_Pal0ad_1n_Python}
```

得到flag

结语

学到新的隐写

Stegosaurus