suctf逆向部分

转载

weixin_30449453 于 2018-06-23 13:46:00 发布 93 收藏
文章标签： python
原文链接：http://www.cnblogs.com/kk328/p/9217060.html

自己真的菜，然后在网上找了一篇分析pyc反编译后的文件然后进行手撸opcode,过程真痛苦
http://www.wooy0ung.me/writeup/2017/10/11/0ctf-quals-2017-py/
names ('ctypes', 'libnum', 'n2s', 's2n', 'binascii', 'b', 'key', 'aaaa', 'aa', 'aaaaa', 'aaa', 'aaaaaa', '__name__')从这我们看到程序的大概函数和变量
ctypes libnum n2s s2n binascii b key aaaa aa aaaaa aaa aaaaaa __name__
查找了一下发现 ctypes 是python 访问c 的库
连接http://python3-cookbook.readthedocs.io/zh_CN/latest/c15/p01_access_ccode_using_ctypes.html
libnum 类似 用法参考以下链接
http://www.cnblogs.com/pcat/p/7225782.html
google 了一下发现这个
Very Hard RSA
http://bestwing.me/2016/09/10/Common%20types%20of%20RSA/
基本还原前四个的代码了
import ctypes
from libnum import n2s,s2n
至于binasciipython内置模块我这里不做阐述
然后就是 b key aaaa aa aaaaa aaa aaaaaa '__name__'
开始猜测大概函数是这样
import ctypes
from libnum import n2s,s2n

```
def b():
    ...
def key():
    ...
def aaaa():
    ...
def aa():
    ...
def aaaaa():
    ...
def aaa():
    ...
def aaaaaa():
    ...
def main():
    ...
if '__name__==main()'
    main()
```

但是再回去分析发现导出都在传key所以key几乎不可能是一个函数而且b只在a.py处出现了一次，推测b可能是一个局部变量

现在有如下结构

```
import ctypes
from libnum import n2s,s2n

def aaaa():
    ...
def aa():
    ...
def aaaaa():
    ...
def aaa():
    ...
def aaaaaa():
    ...
def main():
    ...
if '__name__==main()'
    main()
```

这时候我们通过 names vernames和name 进行还原

```
import ctypes
from libnum import n2s,s2n
```

```
def aaaa():
    a=lambda a:b.hexhexlify(a)

def aa():
    a=cdll.LoadLibrary('./a') #https://blog.csdn.net/linda1000/article/details/12623527
def aaaaa():
    s2n(a)
def aaa():
    a=cdll.LoadLibrary('./a')
def aaaaaa():
    aaa(aaaa(key))
def main():
    aaaaaa()
if '__name__==main()'
    main()
```

发现似乎含有bug,使得freevars段还没使用，怎么办呢google http://kdr2.com/tech/main/1012-pyc-format.html

发现这是嵌套函数使用的，好了们明白了

```
import ctypes
from libnum import n2s,s2n
key=***

def aaaa(key):
    a=lambda a:b.hexhexlify(a)
    return ''.join(a[i] for i in key)
def aa(key):
    a=cdll.LoadLibrary('./a') #https://blog.csdn.net/linda1000/article/details/12623527
    a(key)
def aaaaa(a):
    s2n(a)
def aaa(key):
    a=cdll.LoadLibrary('./a')
    a(key)
def aaaaaa():
    aaa(aaaa(key))
def main():
    aaaaaa()
if '__name__==main()'
    main()
```

程序逻辑到这里差不多清晰了,但是b还有点模糊猜测是import模块引起的，于是在修改

```
import ctypes
from libnum import n2s,s2n
import binascii as b
key=***
```

```
def aaaa(key):
    a=lambda a:b.hexhexlify(a)
    return ''.join(a[i] for i in key)
def aa(key):
    a=cdll.LoadLibrary('./a') #https://blog.csdn.net/linda1000/article/details/12623527
    a(key)
def aaaaa(a):
    s2n(a)
def aaa(key):
    a=cdll.LoadLibrary('./a')
    a(key)
def aaaaaa():
    aaa(aaaa(key))
def main():
    aaaaaa()
if '__name__==main()'
    main()
```

这里基本就复现完成，下面我们在进行解密即可，参考大牛的技术进行后面的解密即可

```
void decrypt(char *k){
    FILE *fp1, *fp2;
    unsigned char key[256] = {0x00};
    unsigned char sbox[256] = {0x00};
    fp1 = fopen("code.txt","r");
    fp2 = fopen("decode.txt","w");
    DataEncrypt(k, key, sbox, fp1, fp2);
}
```

```
extern "C"
{
    void a(char *k){
        encrypt(k);
    }
    void aa(char *k){
        decrypt(k);
    }
}
```

解密时python调用c函数进行解密

```
from ctypes import *
from libnum import n2s,s2n
import binascii as b
#key="20182018"
def aaaa(key):
    a=lambda a:b.hexlify(a)
    return "".join(a(i) for i in key)
def aa(key): #jia mi
    a=cdll.LoadLibrary("./a").a
    a(key)
def aaaaa(a):
    return s2n(a)
def aaa(key): #jie mi
    a=cdll.LoadLibrary("./a").aa
    a(key)
def brup_key():
    i=20182000
    while i<100000000:
        aaa(aaaa(str(i)))
        data=open("flag.txt","r").read()
        if "SUCTF" in data:
            print i
            break
        i=i+1
def aaaaaa():
    # aa(aaaa(key))#jia mi
    # aaa(aaaa(key)) #jie mi
    brup_key()
if __name__=="__main__":
    aaaaaa()
```

key为20182018

参考文章安全客suctfwp链接：https://www.anquanke.com/post/id/146419

转载于:https://www.cnblogs.com/kk328/p/9217060.html