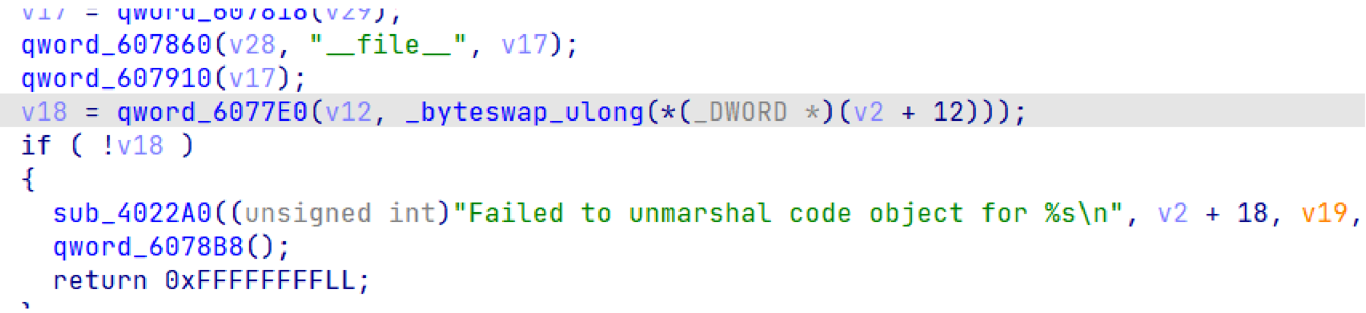
【2020XCTF/华为杯】PYPY WriteUp

原创

古月浪子 于 2020-12-24 17:41:29 发布 390 收藏

文章标签： CTF

题目是一个PYPY打包的elf程序，使用pygame写了一个贪吃蛇游戏，现有的pyinstaller的解包脚本好像不太好使，那么自己动手



```
v17 = qword_607818(v29);
qword_607860(v28, "__file__", v17);
qword_607910(v17);
v18 = qword_6077E0(v12, _byteswap_ulong(*(_DWORD *)(v2 + 12)));
if ( !v18 )
{
  sub_4022A0((unsigned int)"Failed to unmarshal code object for %s\n", v2 + 18, v19,
  qword_6078B8();
  return 0xFFFFFFFFLL;
}
```

根据输出报错信息定位代码，这个qword_6077E0应该是反序列化pyc代码的函数，gdb断在这里，即可拿到对应的pyc文件

这个函数会被执行4次，第4次是我们期望的代码，第四次断下后使用 dump memory ./dmp $rdi $rdi+$rsi 拿到文件，补上前8个字节的文件头以后，加上.pyc后缀然后使用uncompyle6反编译，得到源码

```
DEFAULT_KEY = 'Yó\x02Ã%\x9a\x820\x0b»%\x7f~;ÒÜ'


def rc4(msg, key=DEFAULT_KEY, skip=1024):
    barray = bytearray([i for i in range(256)])
    curr = 0
    for i in range(256):
        curr = (curr + barray[i] + ord(key[(i % len(key))])) % 256
        t = barray[i]
        barray[i] = barray[curr]
        barray[curr] = t
    else:
        curr = 0
        another = 0
        blist = []
        if skip > 0:
            for i in range(skip):
                curr = (curr + 1) % 256
                another = (another + barray[curr]) % 256
                barray[curr], barray[another] = barray[another], barray[curr]

        for j in msg:
            curr = (curr + 1) % 256
            another = (another + barray[curr]) % 256
            barray[curr], barray[another] = barray[another], barray[curr]
            t = barray[((barray[curr] + barray[another]) % 256)]
            blist.append(chr(ord(j) ^ t))
        else:
            return ''.join(blist)


def func():
    t = rc4('flag{this is a fake flag}')
    if t.encode(
            'utf-8').hex() == '275b39c381c28b701ac3972338456022c2ba06c3b04f5501471c47c38ac380c29b72c3b5c38a7ec2a5c2a0':
        return 'YOU WIN'
    return 'YOU LOSE'
```

代码被混淆了，我重命名了一下变量，提取了关键部分贴出来

发现是rc4加密，给了key和加密后的值，当你得到51922968585348276285304963292200096分以后，就会输出YOU LOSE，如果想得到YOU WIN，则需要让flag满足上面的条件

和异或类似，rc4再加密一次就还原了

解密脚本：

```python
import binascii

msg = binascii.a2b_hex('275b39c381c28b701ac3972338456022c2ba06c3b04f5501471c47c38ac380c29b72c3b5c38a7ec2a5c2a0')
key = 'Yó\x02Ã%\x9a\x820\x0b»%\x7f~;ÒÜ'


def rc4(msg, skip=1024):
    barray = bytearray([i for i in range(256)])
    curr = 0
    for i in range(256):
        curr = (curr + barray[i] + ord(key[(i % len(key))])) % 256
        t = barray[i]
        barray[i] = barray[curr]
        barray[curr] = t
    else:
        curr = 0
        another = 0
        blist = []
        if skip > 0:
            for i in range(skip):
                curr = (curr + 1) % 256
                another = (another + barray[curr]) % 256
                barray[curr], barray[another] = barray[another], barray[curr]

        for j in msg:
            curr = (curr + 1) % 256
            another = (another + barray[curr]) % 256
            barray[curr], barray[another] = barray[another], barray[curr]
            t = barray[((barray[curr] + barray[another]) % 256)]
            blist.append(chr(ord(j) ^ t))
        else:
            return ''.join(blist)


print(rc4(str(msg, 'utf8')))
```

> flag{snake_bao_is_really_lucky}