deebugging

发表于 2021-01-15 分类于 Challenge ， 2020 ， 安洵杯 ， Reverse
Challenge | 2020 | 安洵杯 | Reverse | deebugging

点击此处获得更好的阅读体验

WriteUp来源

https://xz.aliyun.com/t/8582

题目考点

解题思路

ida载入，从字符定位到一个假的流程，但可以从base64字符串上面的数据表定位真实流程的一个分支再不断往前找引用来到main函数：

程序创建了子进程，父进程控制修改子进程的eip为指定函数即我们程序的关键函数。这也是我们下断后程序不会断下的原因（自我创建反调试，这里不展开了）。可通过附加子进程调试。

来到关键函数，首先创建了一个线程函数，对flag{i_am_not_right_but_i_believe_you_can_find_out_what_happened}字符进行快速排序后对程序中的2个码表进行异或解密。

附加调试程序起来：

由于附加调试后是上面解码表已经完成后的位置，所以此时直接使用Findcrypt插件可以找到程序使用了加密算法 blowfish，key :who_am_i 直接看blowfish加密的流程还是比较好辨认的。

来到vm部分，注意每次的操作码都有&ff，所以opcode有大量垃圾数据。可以直接在idapthon中打印出所有opcode看看程序操作顺序。

把关键数据跟随到dump窗口调试时，注意一下即可看到操作1把blowfish加密后的数据转化为了字符串。

剩下自己多跟下，可以发现重复的操作很多，再分析下函数功能即可。

*开始初始化了一个链栈和链队列，然后把上面加密了字符串进行依次压栈，再出栈起到倒序的作用，之后把倒序后的字符串进行入队，入队元素个数%4 == 0 时通过一个函数获取当前队列队列中的元素个数n，且从opcode表中取出一个数据data，然后把本次入队的四个元素作为一个整型数据ans，做操作ans -= n*data，最后就是从opcode表中依次取出最后用来比较的数据，四个字节做一个比较，相等的话count++，因为64个字符有16组，所以最后比较count == 16即可。*

idapython提取出用来加密的数据：

提取出最后用来比较的数据:

解密:

```
import struct
from Crypto.Cipher import Blowfish
import codecs

a = [49, 102, 49, 53, 53, 53, 101, 49, 50, 99, 56, 98, 48, 51, 54, 57]

enc = [246, 50, 99, 53, 148, 55, 101, 54, 175, 55, 53, 56, 135, 104, 49, 57, 84, 57, 48, 101, 89, 59, 101, 97, 64, 111, 57, 51, 130, 108, 56, 57, 64, 104, 56, 49, 218, 63,
enc = bytes(enc)
ans = []
ans1 = b''

for i in range(0, 64, 4):
    ans += list(struct.unpack("i", enc[i:i+4]))

for i in range(16):
    ans[i] -=  a[i]*(i+1)*4
    ans1 += struct.pack("i", ans[i])
ans1  = ans1[::-1]
key = "who_am_i"

def Decrypt(enc,key):
        key=key.encode("utf-8")
        #enc=enc.encode("utf-8")
        cl = Blowfish.new(key, Blowfish.MODE_ECB)
        ciphertext = codecs.decode(enc, 'hex_codec')
        code = cl.decrypt(ciphertext)
        return code
flag = Decrypt(ans1, key)
print(flag)
```

Flag

```
d0g3{0f67ca5b9e47c7488309cc021d}
```

- ***本文作者：*** *CTFHub*
- ***本文链接：*** *https://writeup.ctfhub.com/Challenge/2020/安洵杯/Reverse/5C5TYBy4tmAGhJsNmezfZW.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 # 2020 # Reverse # 安洵杯
anxun3
ez_android