

图片隐写linux,CTF-图片隐写-PNG图片修改宽高值的py3爆破

转载

张之光 于 2021-05-14 02:42:20 发布 204 收藏

文章标签: [图片隐写linux](#)

今天拿到一个png图片隐写的题。

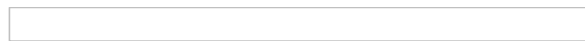
对于这种windows下可以打开, linux下无法打开的png文件, 当然是第一时间怀疑宽和高做了手脚。

爆破crc校验所需要了解到的PNG文件头知识

- (固定)八个字节89 50 4E 47 0D 0A 1A 0A为png的文件头
- (固定)四个字节00 00 00 0D(即为十进制的13)代表数据块的长度为13
- (固定)四个字节49 48 44 52(即为ASCII码的IHDR)是文件头数据块的标示(IDCH)
- (可变)13位数据块(IHDR)
- 前四个字节代表该图片的宽
- 后四个字节代表该图片的高
- 后五个字节依次为:

Bit depth、ColorType、Compression method、Filter method、Interlace method

- (可变)剩余四字节为该png的CRC检验码, 由从IDCH到IHDR的十七位字节进行crc计算得到。



宽和高做了手脚的位置

但我只有手头抄的别人python2.7的脚本, 放在python3下运行不了(.....)

在不断的调试中发现:

有数值超过了127, ascii不认

binascii在python3下不支持unicode编码, 必须输入字节码, 改成b"就能运行了, 但没办法payload拼接在一起爆破

python3把文本和二进制分开, 字节码不能拼接

用string.encode('utf-8')倒是能运行了, 但是没爆破出来, 仔细一看发现编码会把超过ascii范围的譬如\x4拆成b'\xc3\xb4'

.....

搜了一圈又冷静下来, 觉得字节码不能拼接就不拼接嘛.....走别的路径行不行?

行。

搜到bytearray()是个好东西, 相比bytes字节串, bytearray是可修改的。是的我们知道在python2里可直接str拼接, 但是我决定坚守在python3里(其实就是懒得装), 于是开始了数组赋值的过程。

python3.6下的代码如下

```

import zlib

import struct

crc32key = 0xCBD6DF8A #补上0x, winhex下copy hex value。

data = bytearray(b'\x49\x48\x44\x52\x00\x00\x01\xF4\x00\x00\x01\xF1\x08\x06\x00\x00\x00') #winhex下copy
grep hex。

n = 4095 #理论上0xffffffff,但考虑到屏幕实际/cpu, 0x0fff就差不多了

for w in range(n):#高和宽一起爆破

width = bytearray(struct.pack('>i', w))#q为8字节, i为4字节, h为2字节

for h in range(n):

height = bytearray(struct.pack('>i', h))

for x in range(4):

data[x+4] = width[x]

data[x+8] = height[x]

crc32result = zlib.crc32(data)

if crc32result == crc32key:

print(width,height)

return None

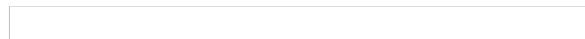
```

一开始用的n是65535, 感觉要花掉一生的时间, 自己手动取消了()

就算这样, 本来也是没有最后这行return的, 我只是把65535换成了4095。跑出结果我花了58s的时间, 丢给老师, 他在虚拟机里只用了3s。

想要挽回自己的尊严()

想要买台新电脑()



心得

今天为了做这道题把png啊struct啊crc啊bytes啊甚至py2到py3的知识全都过了一遍。基础知识还是要扎实啊, 或者出问题的时候再冷静点()

其实写到这里我觉得为什么不直接rb读文件再slice一下而要手动从winhex贴呢, 如此的不neat。但是好困了我要睡觉.....

20170727更新: 一键解决png图片crc隐写的代码

```

import zlib

import struct

#读文件

file = CommonFile+'2.png'

```

```
fr = open(file,'rb').read()
data = bytearray(fr[12:29])
crc32key = eval(str(fr[29:33]).replace("\x","").replace("b","0x").replace("",""))
#crc32key = 0xCBD6DF8A #补上0x, copy hex value
#data = bytearray(b'\x49\x48\x44\x52\x00\x00\x01\xF4\x00\x00\x01\xF1\x08\x06\x00\x00') #hex下copy grep
hex
n = 4095 #理论上0xffffffff,但考虑到屏幕实际, 0x0fff就差不多了
for w in range(n):#高和宽一起爆破
width = bytearray(struct.pack('>i', w))#q为8字节, i为4字节, h为2字节
for h in range(n):
height = bytearray(struct.pack('>i', h))
for x in range(4):
data[x+4] = width[x]
data[x+8] = height[x]
#print(data)
crc32result = zlib.crc32(data)
if crc32result == crc32key:
print(width,height)
#写文件
newpic = bytearray(fr)
for x in range(4):
newpic[x+16] = width[x]
newpic[x+20] = height[x]
fw = open(file+'.png','wb')#保存副本
fw.write(newpic)
fw.close
return None
```