Jarvis OJ - Baby's Crack - Writeup

转载

baikeng3674 于 2017-07-14 10:54:00 发布 73 收藏
原文链接：http://www.cnblogs.com/WangAoBo/p/7168942.html
版权

Jarvis OJ - Baby's Crack - Writeup

> M4x原创，欢迎转载，转载请表明出处

> 这是我第一次用爆破的方法做reverse，值得记录一下

题目：

文件下载

分析：

下载后解压有exe和flag.enc两个文件，初步推测flag.enc是用exe加密后的文件，我们只需要分析exe的加密算法，还原flag.enc的明文即可

exe无壳，拖到IDA中用F5大法，找到关键代码：

```
1 // local variable allocation has failed, the output may be wrong!
2 int __cdecl main(int argc, const char **argv, const char **envp)
3 {
4   int result; // eax@16
5   char Dest; // [sp+20h] [bp-80h]@16
6   FILE *v5; // [sp+88h] [bp-18h]@5
7   FILE *File; // [sp+90h] [bp-10h]@4
8   char v7; // [sp+9Fh] [bp-1h]@6
9   int v8; // [sp+B0h] [bp+10h]@1
10  const char **v9; // [sp+B8h] [bp+18h]@1
11
12  v8 = argc;
13  v9 = argv;
14  main(*(_QWORD *)&argc, argv, envp);
```

```
14  _main( (_QWORD *)&argc, argv, envp);
15  if ( v8 <= 1 )
16  {
17    printf("Usage: %s [FileName]\n", *v9);
18    printf("FileName是待加密的文件");
19    exit(1);
20  }
21  File = fopen(v9[1], "rb+");
22  if ( File )
23  {
24    v5 = fopen("tmp", "wb+");
25    while ( feof(File) == 0 )
26    {
27      v7 = fgetc(File);
28      if ( v7 != -1 && v7 )
29      {
30        if ( v7 > 47 && v7 <= 96 )
31        {
32          v7 += 53;
33        }
34        else if ( v7 <= 46 )
35        {
36          v7 += v7 % 11;
37        }
38        else
39        {
40          v7 -= v7 % 61;
41        }
42        fputc(v7, v5);
43      }
44    }
45    fclose(v5);
46    fclose(File);
47    sprintf(&Dest, "del %s", v9[1]);
48    system(&Dest);
49    sprintf(&Dest, "ren tmp %s", v9[1]);
50    system(&Dest);
51    result = 0;
52  }
53  else
54  {
55    printf("无法打开文件%s\n", v9[1]);
56    result = -1;
57  }
58  return result;
59 }
```

代码的逻辑很简单，exe读取文件中的每一个字符，经过加密存储到flag.enc中，加密的方式有三种：

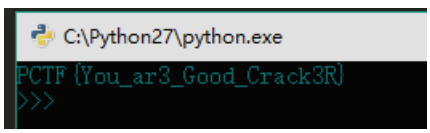
```
if ( v7 != -1 && v7 )
    {
      if ( v7 > 47 && v7 <= 96 )
      {
        v7 += 53;
      }
      else if ( v7 <= 46 )
      {
        v7 += v7 % 11;
      }
      else
      {
        v7 -= v7 % 61;
      }
      fputc(v7, v5);
    }
```

刚开始是想跟进加密的算法推出明文，但后来发现flag.enc中的字符只有52位，而三种加密的算法又很简单，因此完全可以爆破

附上爆破脚本

```python
#coding:utf-8
import string

#将密文转化为ascii码值便于处理，当然没有这一步也是可以的
cipher = 'jeihjiiklwjnk{ljj{kflghhj{ilk{k{kij{ihlgkfkhkwhhjgly'
l = map(ord, cipher)
#l = [106, 101, 105, 104, 106, 105, 105, 107, 108, 119, 106, 110, 107, 123, 108, 106, 106, 123, 107, 102, 108, 103, 104, 104, 106, 123, 105, 108, 107, 123, 107, 123, 107, 105, 106, 123, 105, 104, 108, 103, 107, 102, 107, 104, 107, 119, 104, 104, 106, 103, 108, 121]

#爆破的范围为所有可打印字符
dic = string.printable

#三种加密方式
f1 = lambda x: chr(x - 53)
def f2(x):
    for i in dic:
        if ord(i) + ord(i) % 11 == x:
            return chr(i)

def f3(x):
    for i in dic:
        if ord(i) - ord(i) % 61 == x:
            return chr(i)

#爆破函数，逐位对密文进行爆破
def crack():
    ans = ''
    for i in l:
        try:
            ans += f1(i)
        except:
            pass

        try:
            ans += f2(i)
        except:
            pass

        try:
            ans += f3(i)
        except:
            pass

    return ans

#爆破出的明文是16进制的，还要进行解码
# print crack()
print crack().decode('hex')
```

运行得到flag



转载于:https://www.cnblogs.com/WangAoBo/p/7168942.html