0ctf simpleapk writeup

原创

ling13579 于 2015-04-12 20:44:50 发布 1813 收藏

Java层分析

*Init*中生成*flag.txt*

```
public void init() {
    EasyRe v0 = this;
    String v1 = "flag.txt";
    EasyRe v6 = v0;
    try {
        InputStream v2 = v6.getResources().openRawResource(2131034112);
        byte[] v4 = new byte[v2.available()];
        v2.read(v4);
        FileOutputStream v5 = v0.openFileOutput(v1, 0);
        v5.write(v4);
        v2.close();
        v5.close();
    }
    catch(IOException v6_1) {
        v6_1.printStackTrace();
    }
}
```


按键处理中，将用户输入与*flag.txt*中内容比较。

```
String v8 = "flag.txt";
try {
    FileInputStream v4 = v7.openFileInput(v8);
    byte[] v6 = new byte[v4.available()];
    v4.read(v6);
    String v3 = EncodingUtils.getString(v6, "UTF-8");
}
catch(Exception v7_1) {
    v7_1.printStackTrace();
}

if(v3.equals(v0.et1.getText().toString())) {
```


直接在手机上找到了*flag.txt*文件，得到内容如下:

0ctf{Too_Simple_Sometimes_Naive!!!}

提交，发现不对。

*So*分析：

定位到*init*函数

```
__int64 my_init()
{
  int v0; // r0@1
  __pid_t v1; // r0@3
  __int64 v2; // ST00_8@3

  v0 = j_j_set_logfunction(nullsub_2);
  if ( CheckSig(v0) )
    j_j_exit(0);
  v1 = j_j_getpid();
  LODWORD(v2) = CheckStrace;
  HIDWORD(v2) = CheckPtrace;
  j_j_hook(&unk_5D931004, v1, "libc.", "read");
  return v2;
}
```


猜测应该是*so*库对*read*函数进行了*hook*，导致*java*层读取的数据并不是文件中真正的内容。

另外*checksig*、*checkstrace*、*checkptrace*看这名字都知道不怀好意，*checksig*应该是对签名做判断，*checkstrace*和*checkptrace*应该是反调试的。

不过分析发现*checkstrace*实际是调用*checkptrace*。而*checkptrace*函数的代码肉眼看不出采用了什么反调试技术，只好上调试器看看。

结果调试发现*checkptrace*中间接调用调用的居然是*read*函数，所以基本猜测程序就是用*checkptracehook*了*read*。往下看，发现一个明显的异或操作，对*read*读入的内容进行异或。

```
v15 = _stack_chk_guard;
j_j_memcpy(dest, &unk_5D92F818, 0x23u);
v5 = dword_5D931044;
j_j_hook_precall();
v11 = ((int (__fastcall *)(int, int, int))v5)(v4, v3, v10);// read
v6 = j_j_getpid();
j_j_snprintf(&s, 0xFFu, "/proc/%d/fd/%d", v6, v4);
j_j_memset(&v14, 0, 0x100u);
j_j_readlink(&s, &v14, 0xFFu);
v7 = j_j_strstr(&v14, "/data/data/easyre.sjl.gossip.easyre");
v8 = 0;

if ( v7 )
{
  while ( v8 != v10 )
  {
    *(_BYTE *)(v3 + v8) ^= dest[v8];
    ++v8;
  }
}
j_j_hook_postcall(&unk_5D931004);
```

直接在调试器中获取到异或的值，写了个程序解了一下：

```
charflag[]="0ctf{Too_Simple_Sometimes_Naive!!!}";
charkey[]={0x00,0x00,0x00,0x00,0x00,0x1D,0x1B,0x48,0x2C,0x0C,0x24,0x02,0x02,0x09,0x3A,0x0B,
0x3B,0x0E,0x03,0x3A,0x39,0x0C,0x08,0x11,0x00,0x00,0x1A,0x09,0x0C,0x29,0x20,0x58,
0x44,0x00,0x00};
for(inti = 0; i < strlen(flag); i++)
{
<span style="white-space:pre"> </span>flag[i]^=key[i];
}
printf("%s\n",flag);
```

*flag*为：*0ctf{It's_More_Than_Meets_The_Eye!}*