apk crack writeup - 2017全国大学生信息安全竞赛信安技能赛初赛

原创

luweirao 于 2017-07-15 20:35:06 发布 2131 收藏
分类专栏： 信安CTF 文章标签： CTF android ndk 信息安全 apk

Apk crack Writeup

0x01 反汇编

安卓逆向，首先使用工具得到smali代码和Java代码，一开始用的是Java Decomplier去看的，结果这软件解析的时候有问题，要用别的工具。得到源代码：P_ichunqiu.class、simple.class、wick.class，再使用IDA得到NDK的伪代码：libP_jni.so

0x02 寻找和分析MainActivity

MainActivity应该不难寻找，本题改了个类名，没有用正常的类名。

简单分析得知是通过Simple类的check方法进行校验的，而两个setText的话，看起来貌似是set的空串，但是运行的知是有failed显示的，所以推测在wick.show的时候设置了字符串内容，转到wick可以发现：

的确是加载了库文件，，调用了库中的a函数，而且还存在if，于是对该库进行分析。

0x03 分析libP_jni

简单观察可知，该库的功能仅仅为加载成功和失败时候的两个提示字符串，分别为“success”和“failed”，并无其他价值，故忽略。说明关键点在Java代码中。这个迷惑的还真是666。

0x04 分析check

那就只剩check方法了，结果一打开check方法就惊呆了，547行。

而且还有很多的goto语句，Java并不支持这种用法。先对代码进行目视分析发现，是对输入字符串转换为byte数组，再进行判断，暂时没发现输入条件检查的地方。

check方法中有很多常量，用Python直接还原之后也没有发现有用的信息，但是估计flag可能出自这里，也可能出自用户正确的输入。

0x04 改写Java代码

因为Java不支持goto，所以这个类不能直接运行，而我们当时又不太会Java动态调试（大神是这样做的），既然不会的话就改写为C代码进行调试吧。

新建了一个Xcode工程，为了解决byte到char数组转换问题用了OC的库函数，但是后来发现并没什么用，不转换也可以，同时注意C的话在return那里会有错误，原因是C不支持在中间定义变量，将变量定义移动到开头即可。改写为C代码后可以成功运行并可调试。

0x05 调试C代码

这里用的是AppleLLDB（clang），其实都一样，不过建议用gcc，clang这种Unix/Linux 下的，尽可能接近原始环境，MS VC的话不知道又有什么幺蛾子。

首先在代码中寻找if语句：

发现了一个if(v0<v1)，应该比较关键，下一个断点在那里。再找到return下一个断点。

我们在做题的时候一开始就是单步调试，键盘都快烂了，现在的话直接断在if这里：

此处判断v0和v1的大小关系，这里v1是用户输入的第7位（数组下标6），v0是固定的ASCII码0，字符串截止符。这个通过前面步进分析可以知道。继续步进可以发现，如果if循环条件满足，程序会直接转入return，而且返回失败，非用户可控。这说明字符串的第7位必须是字符串截止符，也就是输入长度必须为6。

注意这里会断两次，其中第一次断，这个条件永远为假，第二次断是上面说的方式。

此处绕过之后反复进入label443，这里一直单步步进的话会发现是往B数组中拷贝了字符串“clo5er”，这个已经怀疑为答案了。

再继续步进会发现又会在if(v0<v1)的地方逐位检查输入字符串是否为刚才的clo5er，如果不是的话直接进入if转入return 0，是才能避开if里面的内容。

最终正确的话程序会返回1，主页面显示success。按格式提交clo5er即为flag。

0x06 总结心得

感觉这道题还是蛮有套路的，就这个check方法的分析非常复杂，当然大神的做法是发现输入长度必须为6之后直接下断点到return处就得到答案。仔细想想也不是没有道理，只是我们经验不足，又想把整个代码debug剖析一遍。所以多耗费了时间，但是还是有收获的，就关于字符串截止符的问题，大一就在学这些，到现在来考察的话，依旧会难住很多人。所以基础还是要牢固啊。