使用frida破解Native层加密

所需设备和环境：

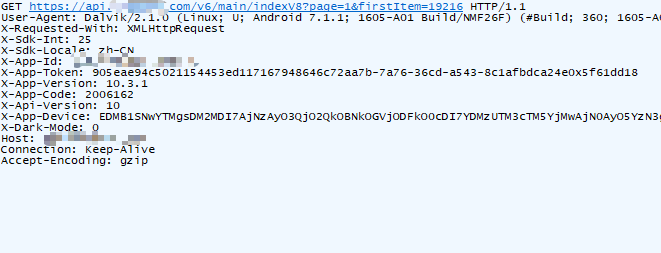
设备：安卓手机(获取root权限)

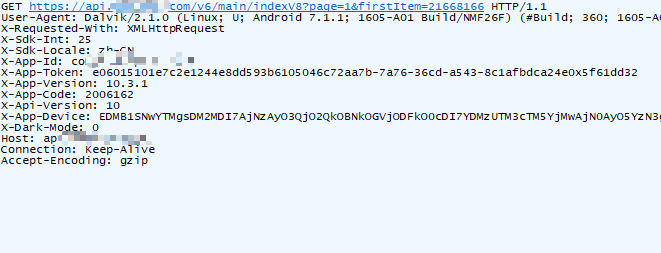
抓包：fiddler + xposed + JustTrustMe

反编译：jadx-gui，ida

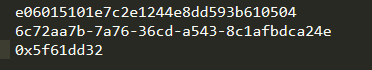
抓包

按照惯例，这里隐去app的名称，开启fiddler抓包后app提示连接不到网络，判断是证书验证，开启xposed框架，再次请求，成功抓到包，连续请求两次来对比参数变化：





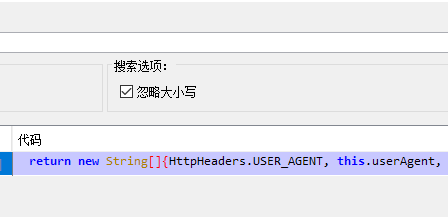
可以看到x-app-token这个参数不一样，其他参数都是固定的，先简单看一下x-app-token的构成：

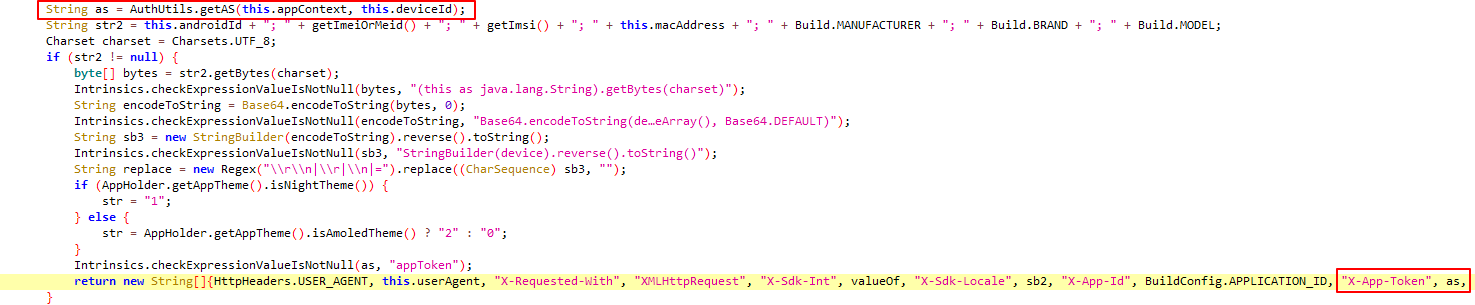


这样做一个换行，有没有发现什么规律？我一眼就看出来第二行是手机的deviceID，因为爬虫写多了，这个字符串经常出现，第一行是32位，应该就是个md5， 第三行是个16进制数，知道了这些，接下来就来反编译，搜索这个关键字。

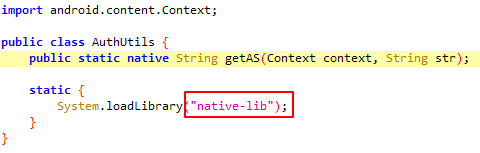
反编译

反编译之前先使用ApkScan-PKID查一下是否有壳，养成好习惯，这个app没有加壳，直接用jadx打开，全局搜索”x-app-token”，只有一处，应该就在这里了：

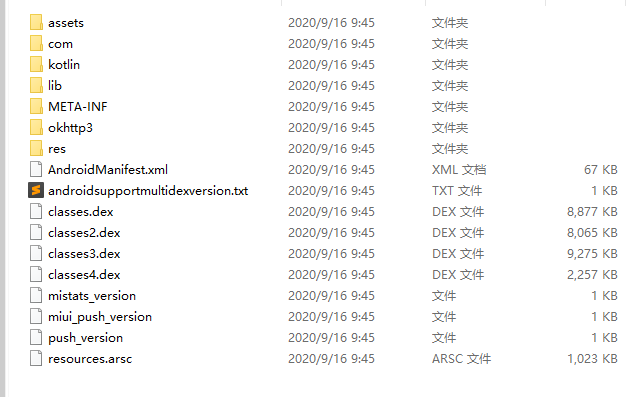




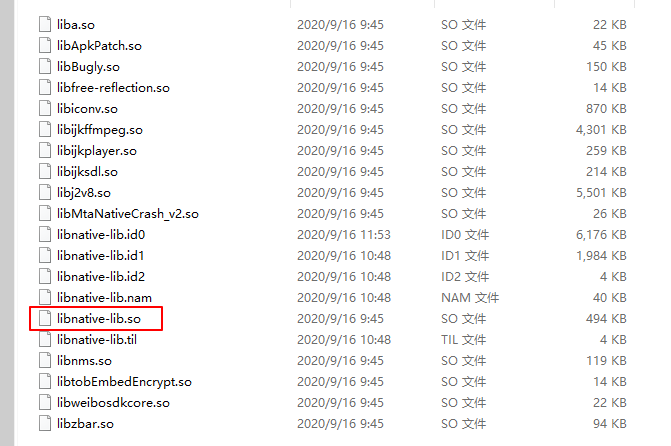
从框中代码发现x-app-token的值就是as，而as是函数getAS生成的，按住鼠标左键，点击getAS进去看看详情：



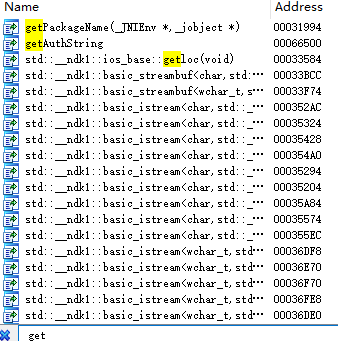
它把函数过程写到native层中去了，那只好去分析这个名为native-lib的so文件了，解压目标apk获取到了如下图这些文件：



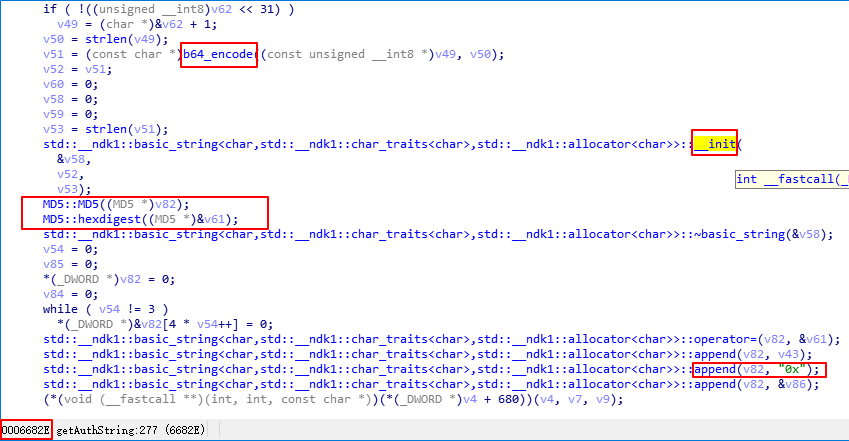
so文件就在这个lib文件夹里：



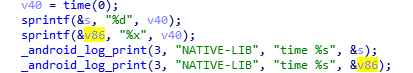
ida中打开这个文件，切换到Export导出函数选项卡，先来直接搜索getAS：



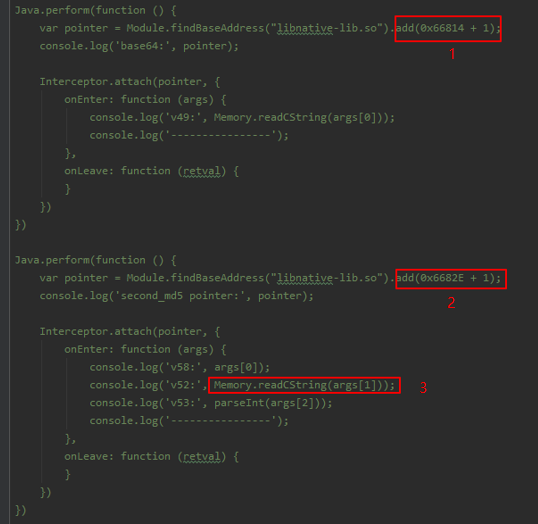
没有搜到getAS，倒是搜到了getAuthString，名字很像，打开看看吧，按F5可以把汇编代码转换为伪c代码，下拉分析一波，看到了md5关键字：



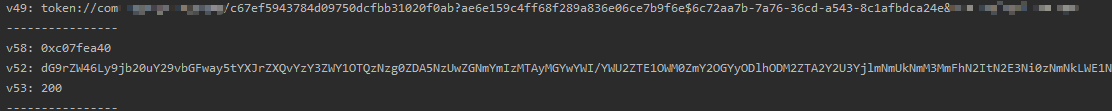
代码有点难懂，连懵带猜v61应该是MD5加密后的结果，然后append似乎是在拼接字符串，看到了“0x”，根据抓包的结果已经知道x\_app\_token的第三行是个16进制数，那应该是这里的v86了，向上看看它具体是什么：



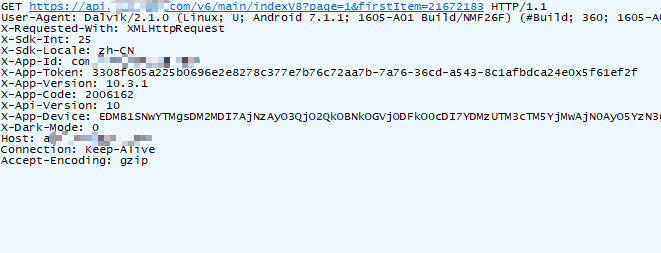
v86存的是v40的值，而v40就是当前的时间戳，那x-app-token第三行应该就是时间戳的16进制数。接下来看看md5的入参为v58,v52,v53，重点来看v52，往上追朔看到其和v51是相等的，而v51是一个未知字符串经过base64编码得到的，先来hook这两个函数，写段脚本：



第1处和2处分别是目标函数的地址，鼠标点到目标函数处，ida左下角会显示，加1是为何，只记得大学时候学过汇编语言讲到段地址，偏移地址，物理地址这些，猜测和这些知识有关，还需研究，这里先这样用，第3处的写法是当这个参数为字符串时的写法，运行一波，看看打印输出：



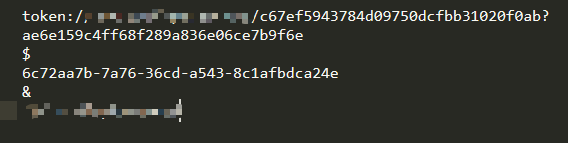
同时也来抓包：



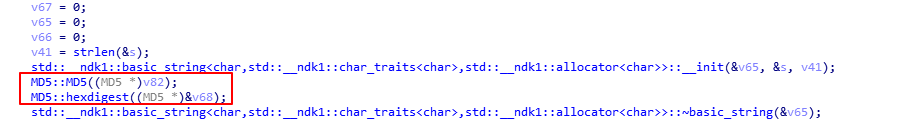
把v52的值放到在线MD5网站中加密看看：



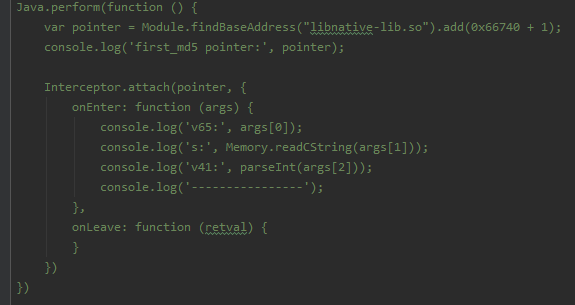
和抓包得到的结果是一样的，那就证明hook点找对了，那么v49就是base64编码之前的值，v49是什么呢，仔细一看，里面也有deviceID，再来分割一下:



第一行通过两次hook，发现他是固定不变的，第二行是改变的，32位，看起来又是个md5，第六行是包名，返回伪c代码，再来分析看看，在前面又看到了MD5加密的地方：



那来hook这个函数，看看入参：



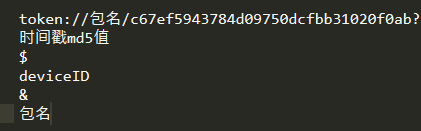
打印的入参数据为：



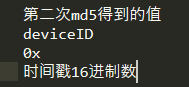
S是一个时间戳，拿到MD5加密网站上验证一下，正好和v49中包含的字符串相同：



至此，逆向就完成了，来总结一下x-app-token的获取过程，先是带有把时间戳进行md5，按下图顺序拼接：



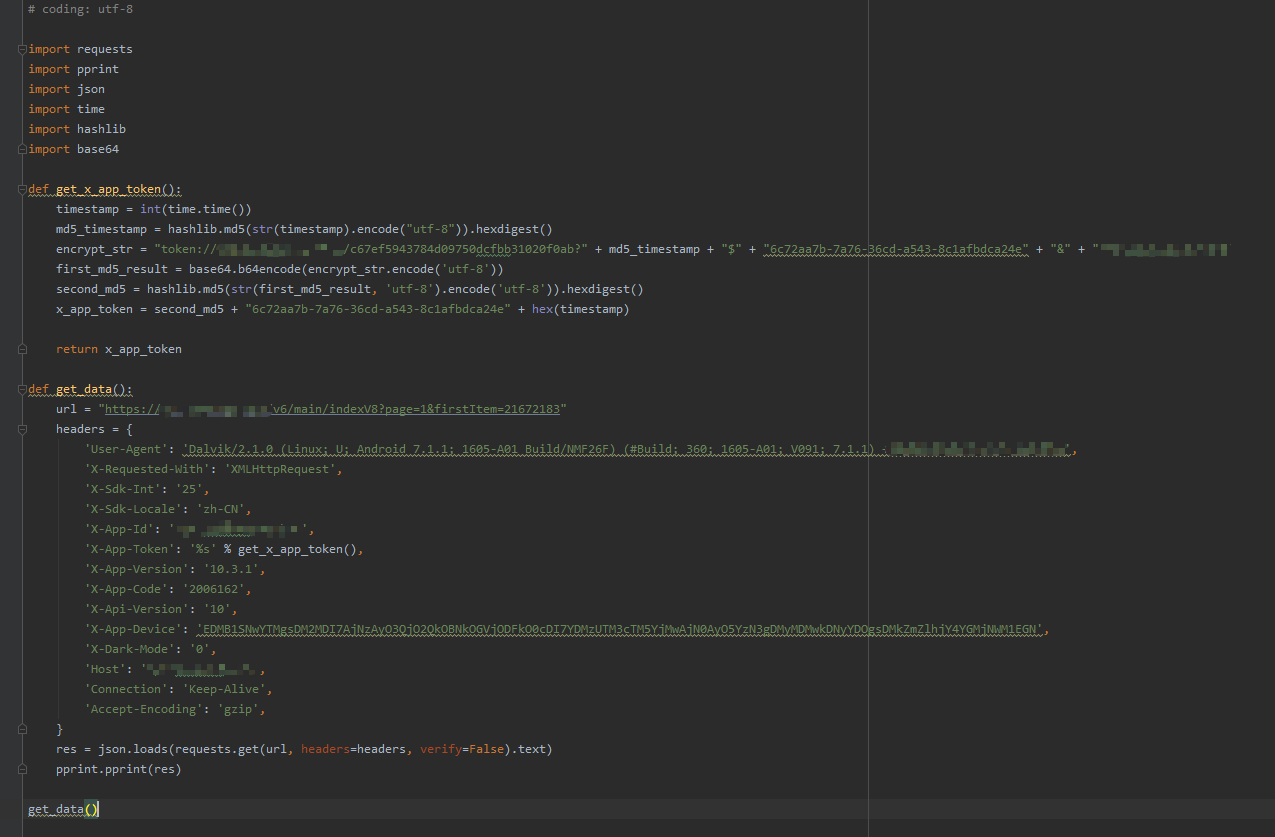
然后把拼好的字符串进行base64编码，最后把编码结果进行MD5，得到x-app-token的开头部分，组成如下：



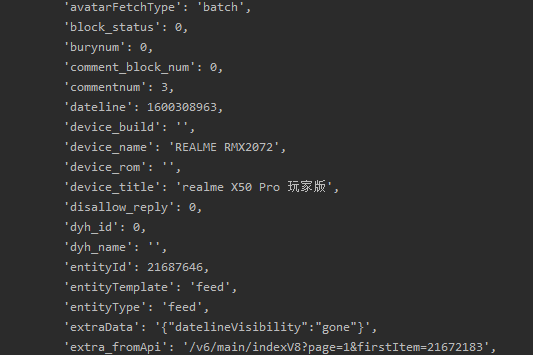
接下来写个脚本来请求数据

请求

代码如下：



运行，成功拿到数据：



总结

今天主要介绍了native层hook的方法，虽然 so文件中的伪c代码要有一定的基础才能看懂，但是先不要怕，分析的过程中可以先大胆假设，有frida非常强大的hook功能，就能一步步验证之前的假设。这两次都是静态分析方法，下次我们再来介绍一下动态调试的方法。

最后，以上内容仅供学习交流。