**CVE–2017–13156 Janus安卓签名漏洞**



0x00 背景介绍

2017年7月31日GuardSquare向Google报告了一个签名漏洞并于当天收到确认。Google本月修复了该漏洞，编号CVE-2017-13156。攻击者可以绕过签名验证机制构造恶意程序更新原有的程序。

0x01 事件概述

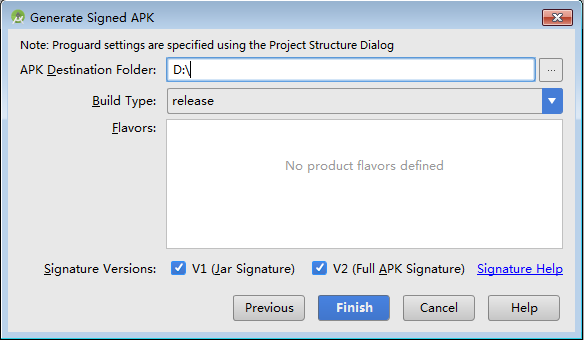
该漏洞产生根源在于将DEX文件和APK文件拼接后校验签名时只校验了文件APK部分，而虚拟机执行时却执行了文件的DEX部分，导致了漏洞的发生。由于这种同时为APK文件和DEX文件的二元性，联想到罗马的二元之神Janus，将该漏洞命名为Janus漏洞。

0x02 事件影响

影响Android5.0-8.0的各个版本和使用安卓V1签名的APK文件。

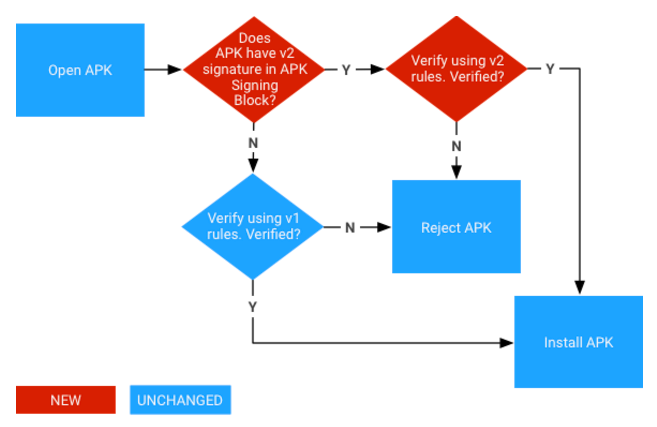
0x03 事件详情

Android支持两种应用签名方案，一种是基于JAR签名的方案（v1方案），另一种是 Android Nougat(7.0)中引入的APK签名方案v2（v2方案）。v1签名不保护APK的某些部分，例如ZIP元数据。APK验证程序需要处理大量不可信（尚未经过验证）数据结构，然后会舍弃不受签名保护的数据。这导致相当大受攻击面。



1. 安卓在4.4中引入了新的执行虚拟机ART，这个虚拟机经过重新的设计，ART允许运行一个raw dex，也就是一个纯粹的dex文件，不需要在外面包装一层zip。而ART的前任DALVIK虚拟机就要求dex必须包装在一个zip内部且名字是classes.dex才能运行。当然ART也支持运行包装在ZIP内部的dex文件，要区别文件是ZIP还是dex，就通过文件头的magic字段进行判断：ZIP文件的开头是‘PK’, 而dex文件的开头是’dex’.

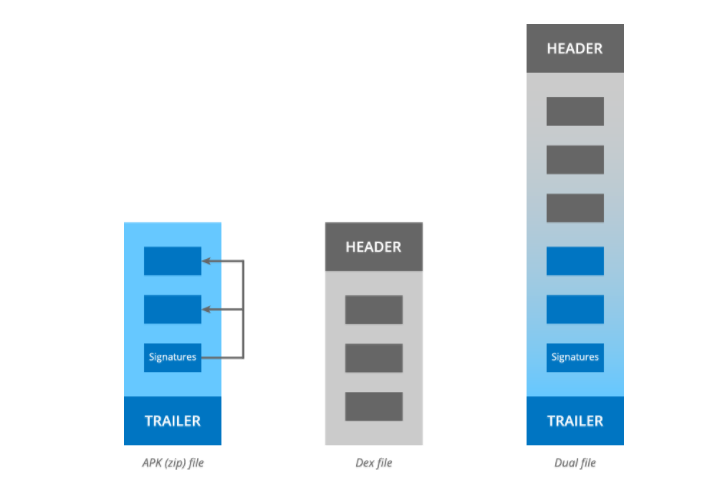
2. ZIP文件的读取方式是通过在文件末尾定位central directory, 然后通过里面的索引定位到各个zip entry，每个entry解压之后都对应一个文件。



**利用过程**

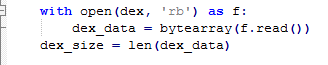
1. 攻击者可以向APK文件的开始位置放置一个攻击的DEX文件A；

2. 安卓系统在安装时用ZIP的读取机制从末尾开始进行文件的读取，读取到了原始的APK内容，且以V1的方式进行校验，认为这个文件正常，没有篡改，APK安装成功；

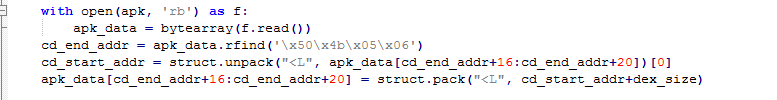
3. 在运行时，Android的ART虚拟机从文件头开始读取，发现是一个DEX文件，直接执行，攻击文件A被最终执行。

我们来看一下已经公布的 POC (<https://github.com/V-E-O/PoC/tree/master/CVE-2017-13156>) 的原理。janus.py接受dex文件和apk文件作为输入，组合起来输出。

读取dex文件：



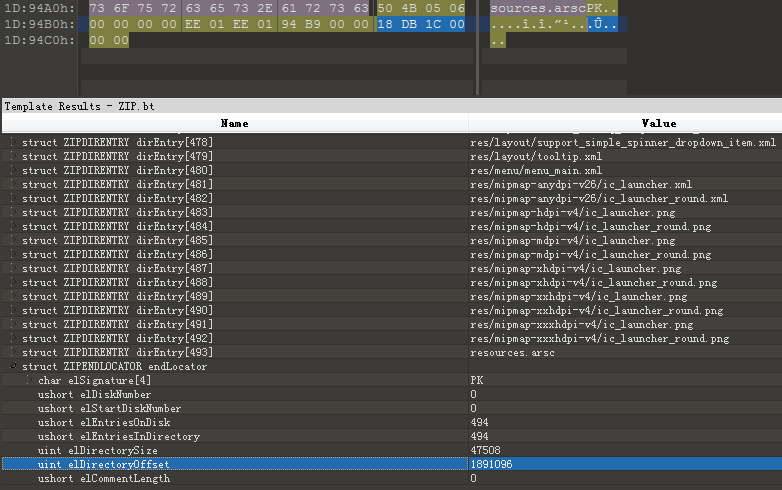
读取apk文件：



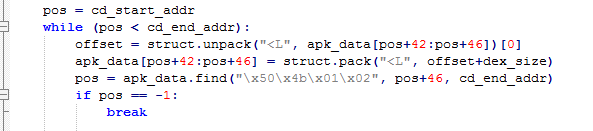
Apk其实就是一个zip。简单地说zip文件格式由文件数据区、中央目录结构和中央目录结束节组成。



其中，中央目录结束节有一个字段保存了中央目录结构的偏移。代码中搜索中央目录结束节的固定结束标记x06054b50定位到中央目录结构的偏移，将其加上dex文件的大小，因为我们要把dex文件插到apk前面。

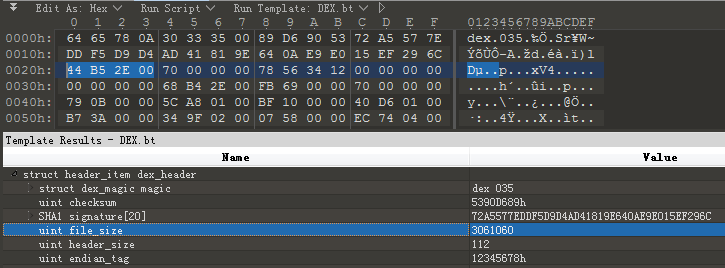


接下来依次更新中央目录结构数组中的deHeaderOffset字段也就是本地文件头的相对位移字段。通过deHeaderOffset字段可以直接获取到对应文件的文件数据区结构的文件偏移，就可以直接获取到对应文件的压缩数据了。同样也是因为dex文件插在前面了所以直接加上dex文件的大小。

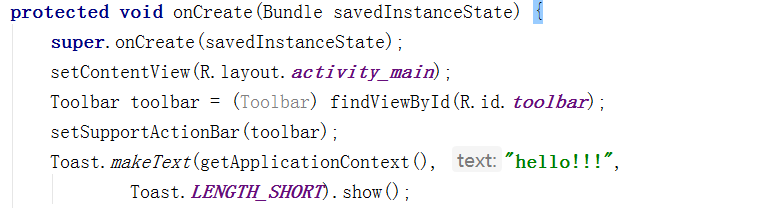


最后更新dex部分的file\_size字段为整个dex+apk的大小，使用alder32算法和SHA1算法更新checksum和signature字段。

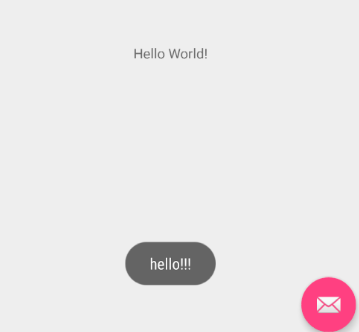
enter image description here



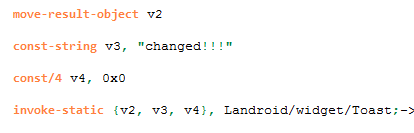
做一个测试。 在APK文件中写一个弹出Hello的toast，同时采用V1签名方案签名：



安装到手机上：



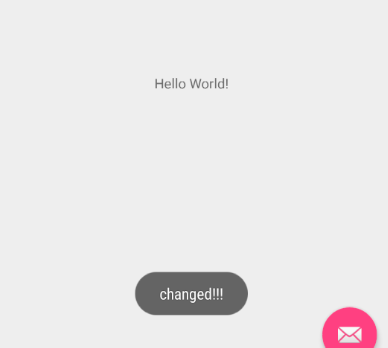
将编译好的apk解压得到dex文件，baksmali.jar反编译dex文件得到smali代码，将hello随便改成另外一个字符串：



用smali.jar回编译成dex文件，使用脚本把dex文件和原来的apk打包生成out.apk：

enter image description here

安装到手机上成功通过了签名校验并且执行了修改的dex中的代码：

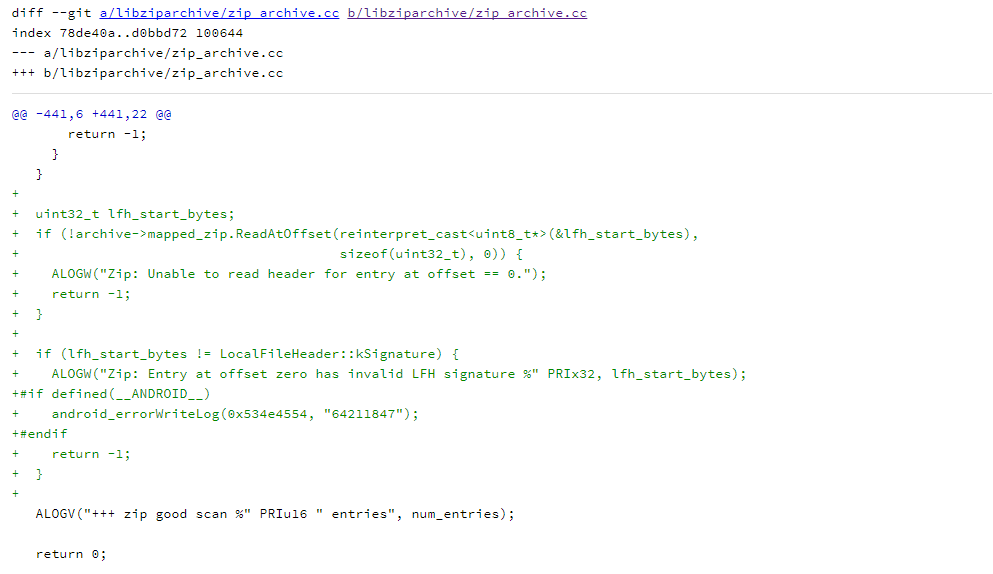


在我android8.0没有打补丁的手机上如果采用了V2签名方案不受该漏洞影响，更新不了原来正常的程序：



补丁分析

补丁非常简单，强制校验了zip的frSignature：



0x04 修复建议

1. 开发者在开发应用程序时勾选V2签名方案；

2. 开发者及时校验App APK文件的开始字节，以确保App未被篡改；

3. 用户应在正规的应用市场下载程序。

0x05 参考文档

<https://source.android.com/security/bulletin/2017-12-01>

<https://source.android.google.cn/security/apksigning/v2?hl=zh-cn>