

Android N native API

孙延宾

4 月 28 日, 2017 年

目 录

1	影响范围	1
2	影响行为	1
3	实现原理	1
4	解决方案	1
5	实用技巧	1
6	参考文献	1
7	附录	2

从 Android N(API 24) 开始, android 限制应用使用 native API, 只允许应用使用公开的 API, 这些 API 会进行 CTS 测试, 从而保证兼容性和稳定性。API 以 “.so” 文件的形式发布。

1 影响范围

只有普通应用受到此规定的约束, 系统应用、HALs, daemon 进程不受影响, 它们依然可以使用非公开库。

2 影响行为

普通应用打开 .so 文件的任意行为均受限制, DT_NEEDED 方式 (编译时引用动态库) dlopen 方式 System.loadLibrary() (本质上使用 dlopen 函数) 以上行为均受该规定约束。

3 实现原理

该限制是在 android 的动态链接器中实现的, 源码在 bionic/linker/ 中, 所以无论是编译期引用, dlopen 还是 loadLibrary() 都无法跳过该规则, 因为它们都是通过 linker 实现的。

4 解决方案

vendor 允许提供额外的公开库,

1. 把需要公开的库文件放到 /vendor/lib 或者 /vendor/lib64 目录中
2. 把库文件的名字添加到 /vendor/qcom/proprietary/common/config/public.libraries.txt 中, 该文件最终会被复制为手机的 /vendor/etc/public.libraries.txt

此外, 也可以将用到的非公开库 (依赖关系比较简单的库, 如 libcutils.so) 复制到 apk 中, 官方推荐将这些库的源码复制到项目的源码树中开发维护。

5 实用技巧

- `readelf --dynamic libExample.so | grep NEEDED` 可以查看依赖哪些动态库
- 在 root 过的手机上, 把普通应用的安装目录 (如 /data/app/com.ybin.demo/) move 到 /system/priv-app/ 中, 即可将普通应用转换为 system 应用。

6 参考文献

- https://source.android.com/devices/tech/config/namespaces_libraries
- <https://android-developers.googleblog.com/2016/06/improving-stability-with-private-cc.html?m=1>
- <https://www.slideshare.net/mobile/BinYang7/linker-namespace-upload>
- <http://forum.cocos.com/t/ndk-android/39547>

7 附录 (public native API, libXXX.so)

1. android
2. c
3. dl
4. jnigraphics
5. log
6. m
7. m_hard
8. stdc++
9. z
10. EGL
11. GLESv1_CM
12. GLESv2
13. GLESv3
14. vulkan
15. OpenSLES
16. OpenMAXAL
17. mediandk
18. atomic

这些公开库在 `platform/system/core/rootdir/etc/public.libraries.android.txt` 中定义，该文件会复制到手机的 `/system/etc/public.libraries.txt` 文件中。注意，该文件会进行 CTS 测试，所以无法修改。此外，NDK 的 `platforms/android-X/<arch>/usr/lib` 目录提供了完整的公开库列表。