FLEXDROID: Enforcing In-App Privilege Separation in Android paper summary

Ruturaj Kiran Vaidya
University of Kansas

1 Summary

Third party libraries violate user privacy, by accessing user sensitive information, without even notifying the user. Current android platform provides security against these libraries, by allowing them to work with the same permissions as their host applications. The authors analyze 100000 apps, to investigate the characteristics of third party libraries. They found that most of these libraries use undocumented permissions, rely on dynamic code and some of them use JNI (which can be a potential threat). "The paper presents 'FLEXDROID', an extension to the android permission system, that gives control to the developers over third party libraries that access user's private information. 'FLEXDROID' provides an interface (as a part of the app manifest), for app developers to specify a set of different permissions granted to each third-party library"[1]. Upon request of user information, it identifies the principal of currently running code using 'inter-process stack inspection', it accepts or denies the request, by adjusting app permissions dynamically, as per the pre-specified permissions in the manifest. Despite of the challenges, authors claimed that 'FLEXDROID' is effective in defense against privacy threats from third-party libraries, by supporting in app privilege separation, resisting against JNI, reflection and multi-threading attacks, and shows reasonable performance overhead.

2 Contribution

The authors proposes 'FLEXDROID' as a defense against third party library threats. It provides in-app privilege separation in android, and supports JNI, reflection and multi-threading. It is the first system that adopts fault isolation using ARM memory domain to sandbox native code in android.

3 Strengths

The paper gives in depth information of the tool implementation, unlike the previous papers I read. The authors analyzed over 100000 apps, which I think is a lot. I liked that 'FLEXDROID' supports JNI, reflection and multi-threading. Also, it shows a sensible overhead (in my opinion).

4 Limitations

As 5 apps are crashed, there are still backward comparability issues making it unreliable. Also, as 'FLEXDROID' gives control to the developer, user data security still depends on the developer (i.e. how responsible is he). Also, they should have used a heavy weight application instead of k-9 app, while benchmarking. The results might have been different (just my opinion). Also, overhead added during launching of service is pretty high (5.22%).

5 Future Work

More work needed on developing JNI sandbox, which provides complete backward compatibility. Also, more work needed on raising awareness of these type of attacks, amongst developers and even users. And, work needed to Establish memory isolation between third-party libraries.

References

[1] Jaebaek Seo, Daehyeok Kim, Donghyun Cho, Taesoo Kimy, Insik Shin *FLEXDROID: Enforcing In-App Privilege Separation in Android*, NDSS, (2016)