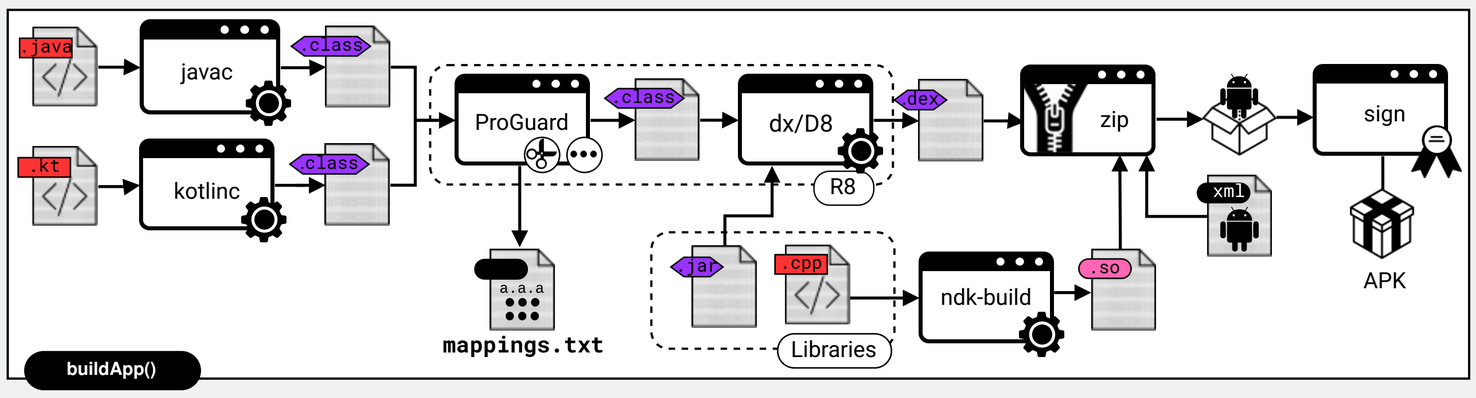
ART-assisted Android App Diffing

Jakob Bleier, Martina Lindorfer
[jakob.bleier, martina.lindorfer]@tuwien.ac.at

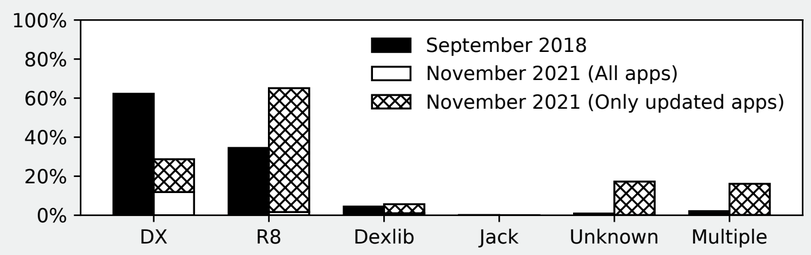
Challenges

R8 is the new default compiler and integrates ProGuard features by default:

- eliminates dead code
- renames classes & methods
- optimizes dalvik code

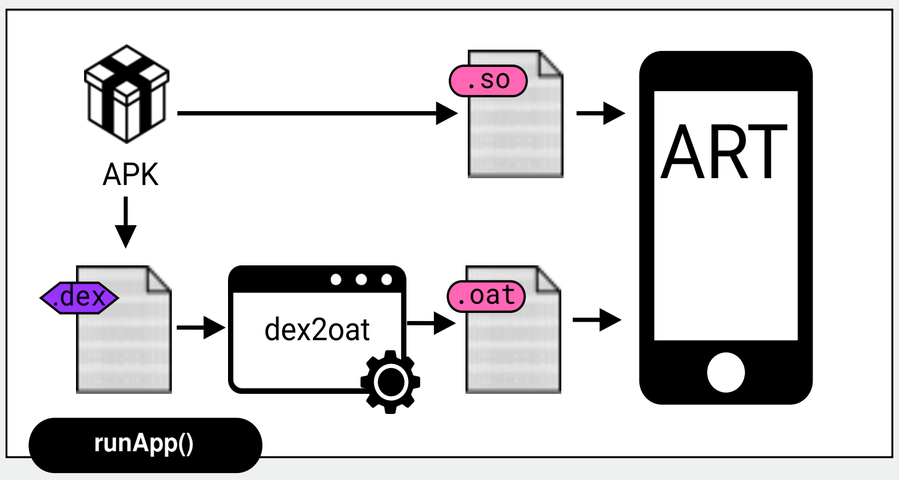


These changes affect analysis tools that assume method signatures are preserved, or that all library code is present in an APK.



Opportunity

The Android Run-Time (ART) compiles Dalvik to binary code before executing it. The compilation occurs just in time, but a full compile can be manually triggered to produce an OAT (Of-Ahead-Time) file, which is also a valid ELF shared object. This can be processed by standard binary analysis tools such as Ghidra, IDA Pro, Binary Ninja, and Angr among others.

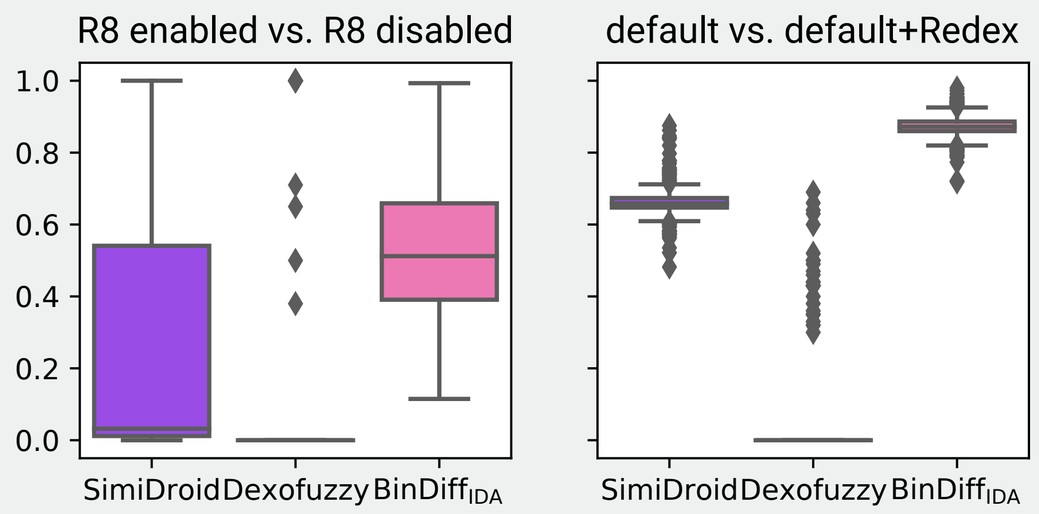


Dataset

- Extending the F-Droid build server, we create a reliable and reproducible ground truth for thousands of apps.
- We can read and adjust build settings, en- or disable R8 settings, and export detailed build information.
- All APKs can be compiled to OATs on Hardware, Emulators or cross-architecturally with the AOSP.
- Additional passes such as Redex and ObfuscAPK are supported.

Results

In the use case of app similarity, $\text{BinDiff}_{\text{IDA}}$ outperforms SimiDroid for detecting apps compiled with different settings. It also provides a score for 99.3% of app pairs, while SimiDroid only produces a result for 65,3% of pairs.



TU WIEN Informatics
S&P Security & Privacy Research Unit
SBA Research