Reverse Engineering Android apps with ACVTool

Aleksandr Pilgun, PhD
Independent Researcher

2024-10-25 Luxembourg

Disclaimer

This presentation is for **educational purposes only**.

The tools, techniques, and methodologies demonstrated are intended for learning and research.

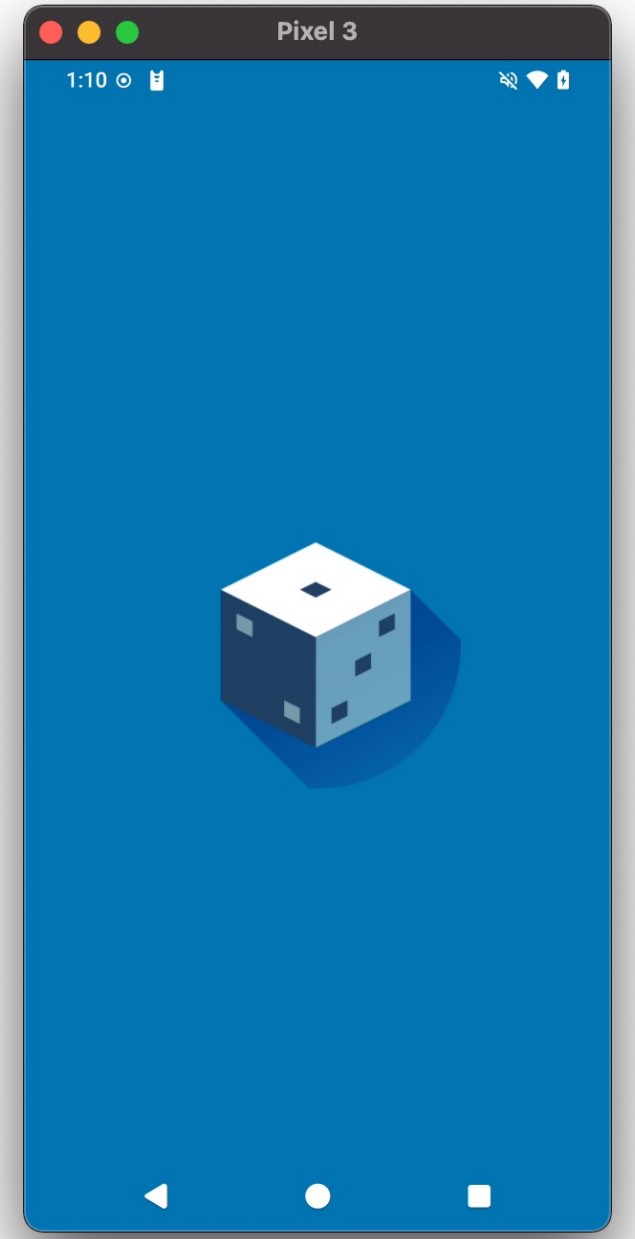
Any similarities between decompiled or disassembled code shown here and proprietary code are purely coincidental.

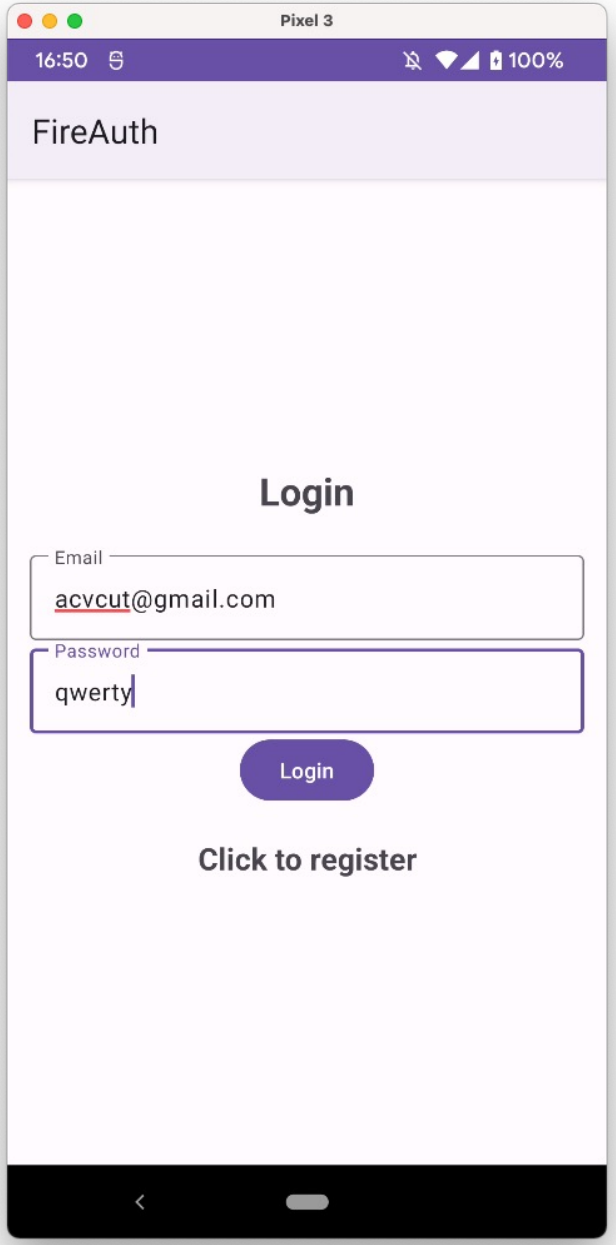
I do not endorse or promote any illegal activities, and the content of this talk is shared with the intent of fostering knowledge within legal and ethical boundaries.

Outline

- Android apps & Reverse Engineering
- ACVTool
- ACV Shrinking
- Examples

Android app


Pixel 3
1:10


Pixel 3
16:50
100%
FireAuth
Login
Email
acvcut@gmail.com
Password
qwerty
Login
Click to register

Android Package

Single DEX APK

base.apk
- assets/
- libs/
- res/
- **classes.dex**
- resources.arsc
- AndroidManifest.xml

Multi-DEX

base.apk
- assets/
- libs/
- res/
- **classes.dex**
- **classes2.dex**
- **classes3.dex**
- resources.arsc
- AndroidManifest.xml

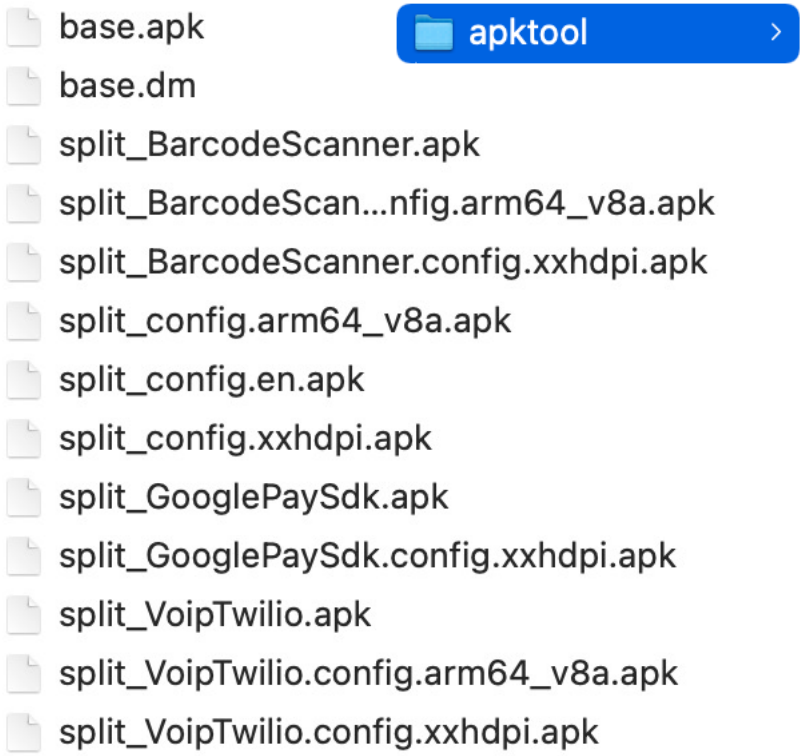
Multi-DEX Multi-APK

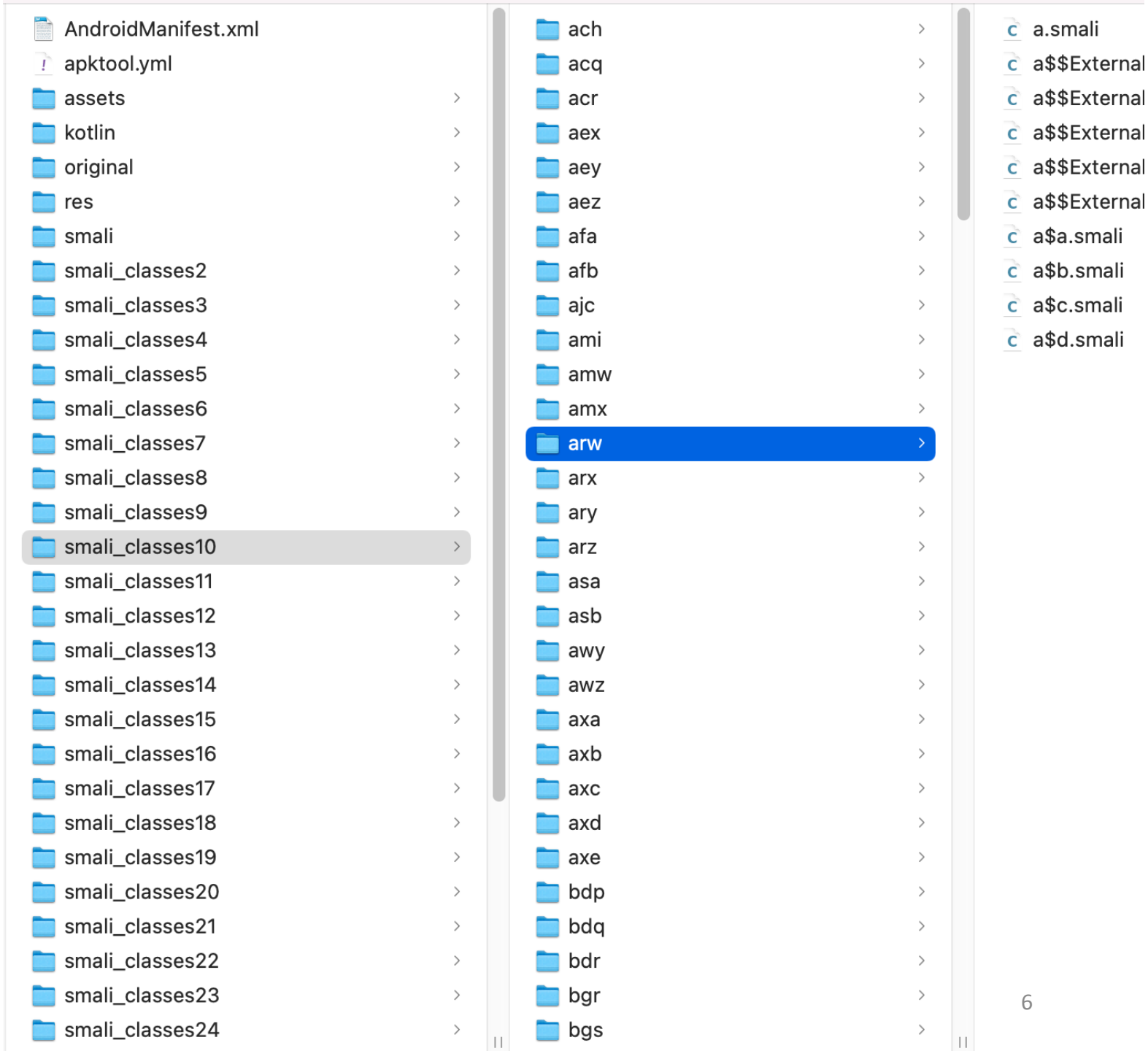
base.apk
- assets/
- libs/
- res/
- **classes.dex**
- **classes2.dex**
- **classes3.dex**
- resources.arsc
- AndroidManifest.xml

split_config.arm64_v8a.apk
split_config.xxhdpi.apk

> adb shell pm list packages -3 -f
> adb **pull** /data/app/~~p1C08tBg3vTKWFOLnqM7fw==/com.package-e8F14npRmhydaMLVb7Xilg==/

Modern Android app





Android Tools Zoo

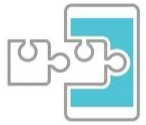
Android SDK

JADX, Apktool, baksmali

Ghidra, IDA Pro, JEB, Radare2

BurpSuite, Mitmproxy, Charles

Xposed Framework, Magisk, Custom ROMs



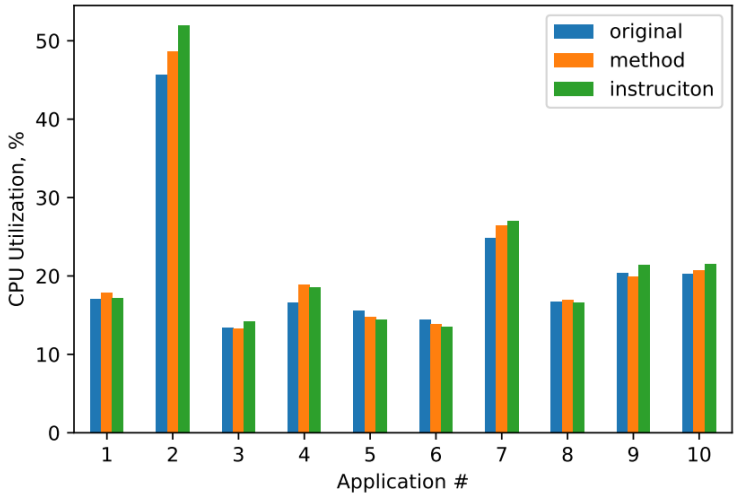
ACVTool

ACVTool

Instruments the Android app

Registers executed instructions

Highlights executed instructions



(a) Average CPU utilization measured in 10 applications.

Paper: Fine-grained code coverage measurement in automated black-box android testing
https://satoss.uni.lu/members/sjouke/papers/PGZDKM20.pdf

Any time
Since 2024
Since 2023
Since 2020
Custom range...

Sort by relevance

Create

AndroLog: Android Instrumentation and Code Coverage Analysis

J Samhi, A Zeller - Companion Proceedings of the 32nd ACM …, 2024 - dl.acm.org

… As demonstrated in this paper, AndroLog can instrument up to 98% of recent Android apps compared to existing tools with 79% and 48% respectively for COSMO and **ACVTool**. …

☆ Save Cite Cited by 1 Related articles All 3 versions

WallMauer: Robust Code Coverage Instrumentation for Android Apps

…n.org

…**ol** and

…cess rates […

(based on JaCoCo [11]), and ACVTool [17]. These tools, however, come with a range of limitations and challenges. Firstly, they are significantly outdated, with no maintenance for 8, 4, and 4 years, respectively for … updates rer…

MiniMon: Minimizing Android Applications with Intelligent Monitoring-Based Debloating

J Liu, Z Zhang, X Hu, F Thung, S Maoz, D Gao… - Proceedings of the …, 2024 - dl.acm.org

… **Pilgun** et al. removed the unexecuted instruction during testing [51]. We try to compare MiniMon with **Pilgun**'s … **Pilgun promised** to run their tool on more apps but has not delivered when …

☆ Save Cite Cited by 1 Related articles All 4 versions

… For this analysis, we leveraged **ACVTool** [2] to measure the percentage code covered for each package of an APK. The first phase involved measuring the code coverage while …

☆ Save Cite Related articles All 2 versions

Computer-Vision-Enabled Worker Video Analysis for Motion Amount Quantification

H Iyer, N Macwan, S Guo, H Jeong - arXiv preprint arXiv:2405.13999, 2024 - arxiv.org

The performance of physical workers is significantly influenced by the quantity of their motions. However, monitoring and assessing these motions is challenging due to the complexities …

☆ Save Cite Related articles All 2 versions »

ACVTool 2024

https://github.com/pilgun/acvtool

ACVTool 2.3.2

- Implements **MultiDex**
- replaces Apktool with **acvpatcher**
- selected feature coverage measurement
- experimental **shrinking**

ACVPatcher

> acvpatcher --apkpath base.apk --class ./classes.dex ./classes2.dex

https://github.com/pilgun/acvpatcher

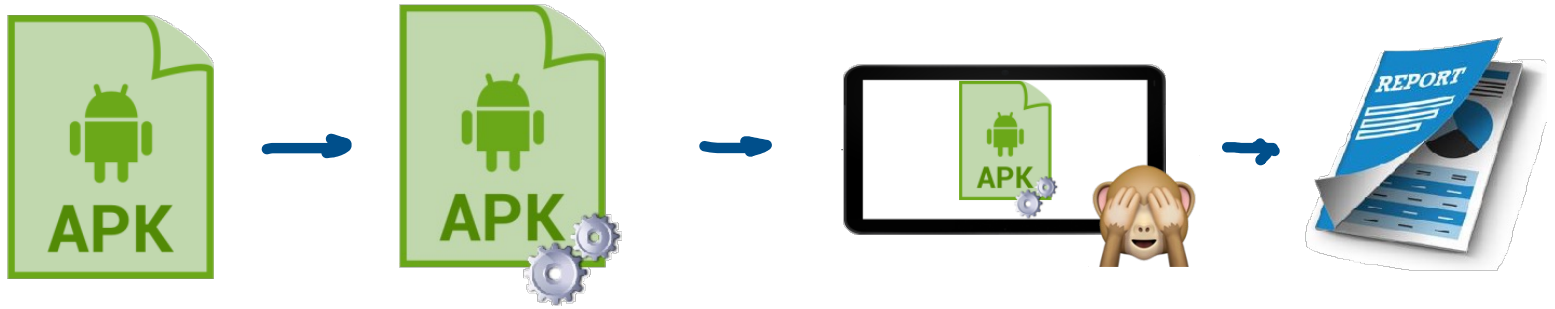
ACVTool Workflow

package_1.pickle
package_2.pickle

instr_package.apk

covered/package_1.pickle
covered/package_2.pickle

coverage_1727615863074_1.ec
coverage_1727615904540_2.ec



> acv **instrument** base.apk

> acv **install** ~/acvtool/acvtool_working_dir/instr_com.package.android.apk

> acv **activate** com.package.android

> acv **snap** com.package.android

> acv **cover-pickles** com.package.android

> acv **report** com.package.android

ACVTool Report

Element	Ratio	Cov.	Missed	Lines	Missed	Methods	Missed	Classes
1		7.56663%	127009	137406	11081	12447	1689	2026
Total	10397 of 137406	7.56663%	127009	137406	11081	12447	1689	2026

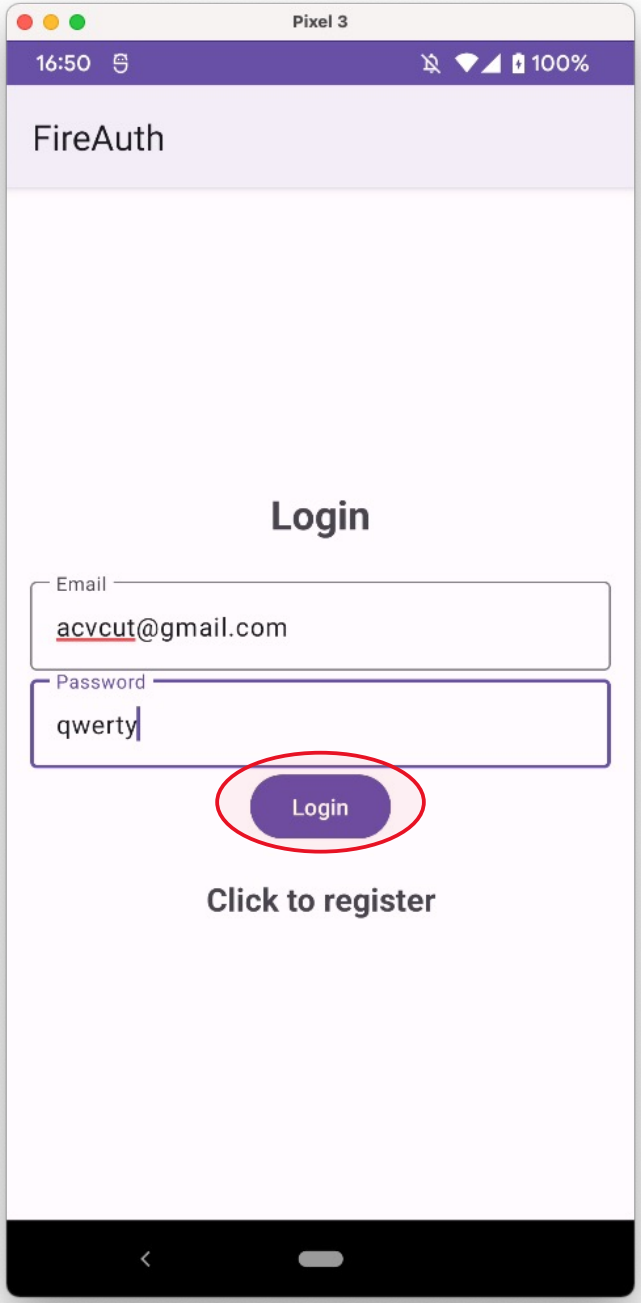
```
.method protected static limitCharSequenceLength(Ljava/lang/CharSequence;)Ljava/lang/CharSequence;
        .locals 2

        if-nez p0, :cond_0
        return-object p0
        :cond_0
        invoke-interface {p0}, Ljava/lang/CharSequence;->length()I
        move-result v0
        const/16 v1, 0x1400
        if-le v0, v1, :cond_1
        const/4 v0, 0x0
        invoke-interface {p0, v0, v1}, Ljava/lang/CharSequence;->subSequence(II)Ljava/lang/CharSequence;
        move-result-object p0
        :cond_1
        return-object p0
.end method
```

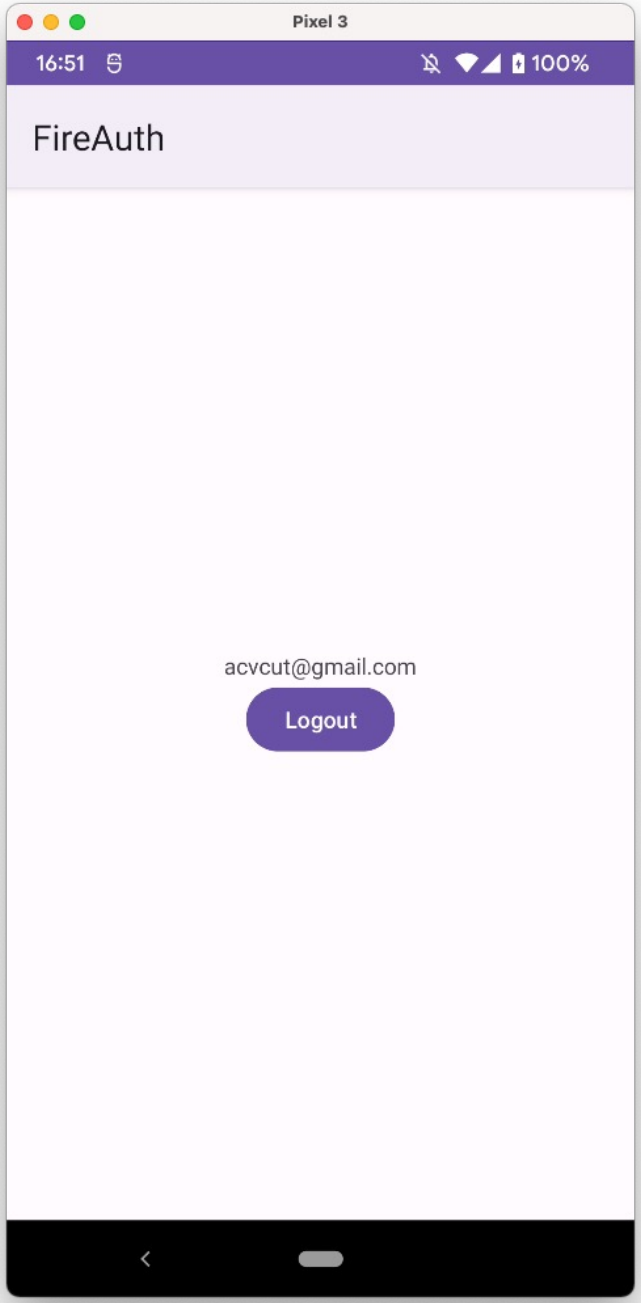
ACV Shrinking

Element	Ratio	Cov.	Missed	Lines	Missed	Methods	Missed	Classes
1		7.56663%	127009	137406	11081	12447	1689	2026
Total	10397 of 137406	7.56663%	127009	137406	11081	12447	1689	2026

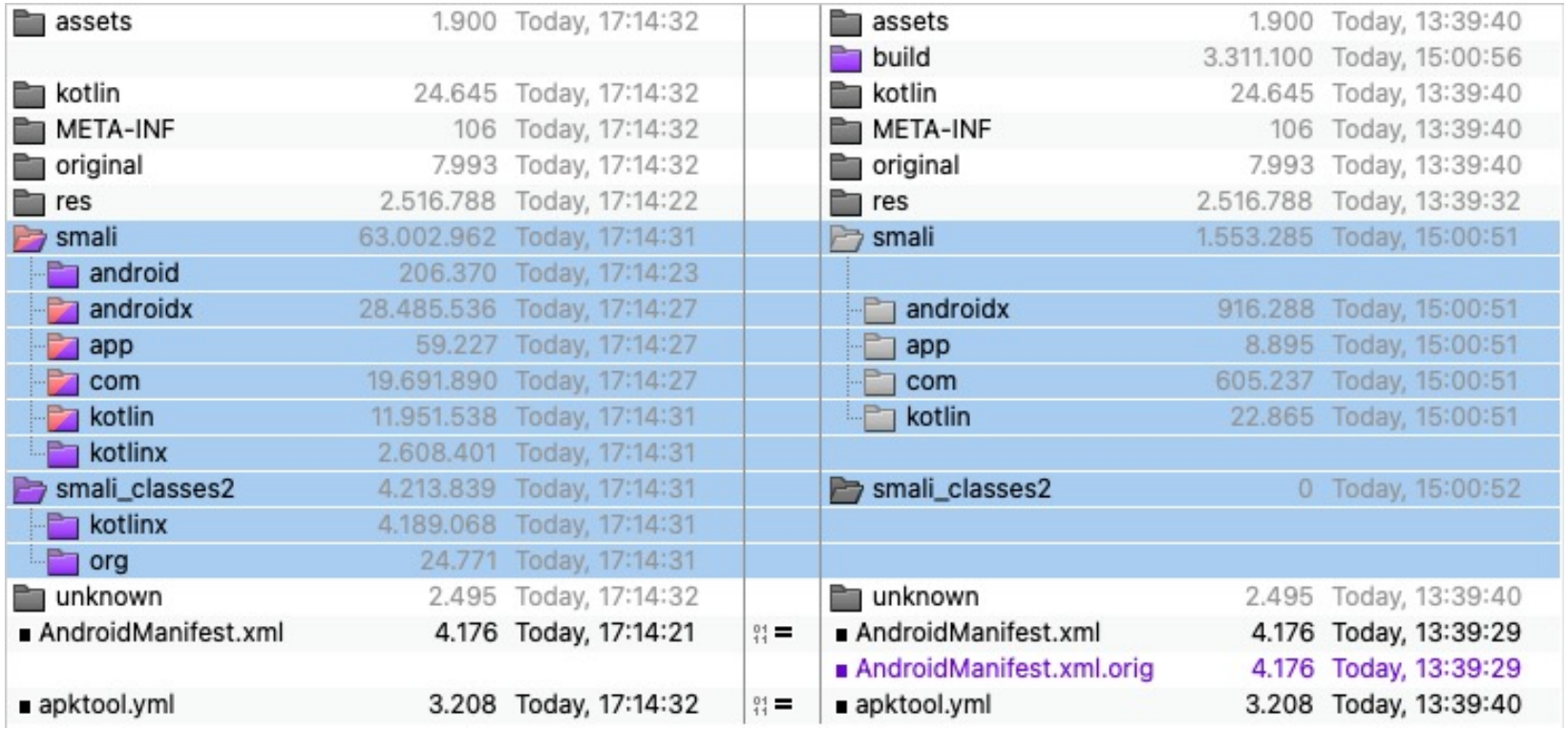
Element	Ratio	Cov.	Missed	Lines	Missed	Methods	Missed	Classes
1		97.52811%	244	9871	11081	12447	1689	2026
Total	9627 of 9871	97.52811%	244	9871	11081	12447	1689	2026

Paper: Don’t Trust Me, Test Me: 100% Code Coverage for a 3rd-party Android App
https://orbilu.uni.lu/bitstream/10993/44480/1/APSEC20_preprint.pdf


Pixel 3
16:50
100%
FireAuth
Login
Email
acvcut@gmail.com
Password
qwerty
Login
Click to register


Pixel 3
16:51
100%
FireAuth
acvcut@gmail.com
Logout

ACV Shrinking



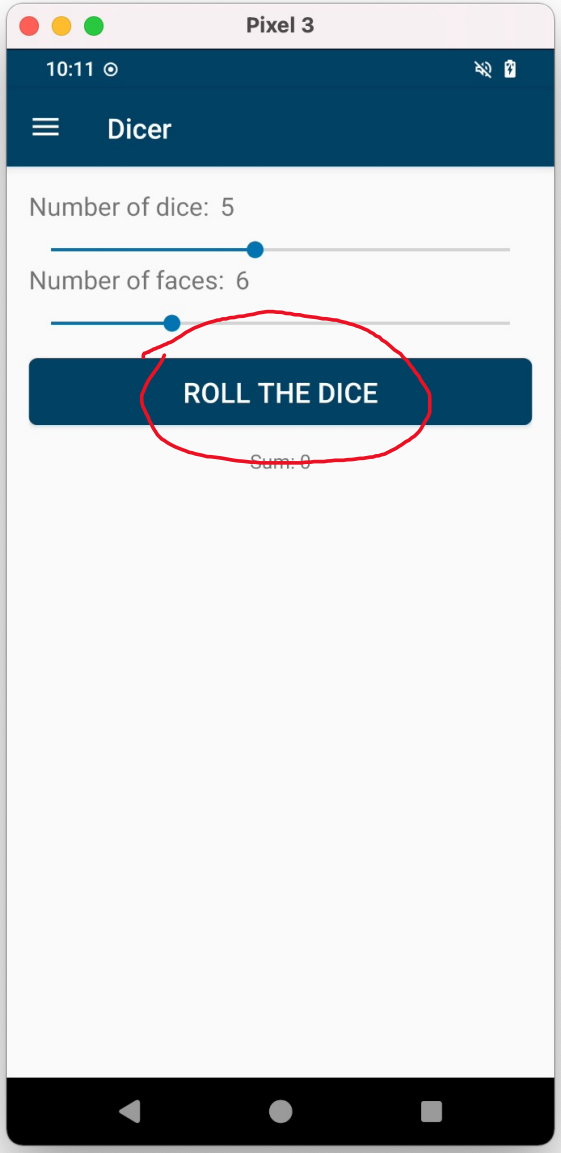
assets	1.900	Today, 17:14:32		assets	1.900	Today, 13:39:40
				build	3.311.100	Today, 15:00:56
kotlin	24.645	Today, 17:14:32		kotlin	24.645	Today, 13:39:40
META-INF	106	Today, 17:14:32		META-INF	106	Today, 13:39:40
original	7.993	Today, 17:14:32		original	7.993	Today, 13:39:40
res	2.516.788	Today, 17:14:22		res	2.516.788	Today, 13:39:32
smali	63.002.962	Today, 17:14:31		smali	1.553.285	Today, 15:00:51
android	206.370	Today, 17:14:23				
androidx	28.485.536	Today, 17:14:27		androidx	916.288	Today, 15:00:51
app	59.227	Today, 17:14:27		app	8.895	Today, 15:00:51
com	19.691.890	Today, 17:14:27		com	605.237	Today, 15:00:51
kotlin	11.951.538	Today, 17:14:31		kotlin	22.865	Today, 15:00:51
kotlinx	2.608.401	Today, 17:14:31				
smali_classes2	4.213.839	Today, 17:14:31		smali_classes2	0	Today, 15:00:52
kotlinx	4.189.068	Today, 17:14:31				
org	24.771	Today, 17:14:31				
unknown	2.495	Today, 17:14:32		unknown	2.495	Today, 13:39:40
AndroidManifest.xml	4.176	Today, 17:14:21	=	AndroidManifest.xml	4.176	Today, 13:39:29
				AndroidManifest.xml.orig	4.176	Today, 13:39:29
apktool.yml	3.208	Today, 17:14:32	=	apktool.yml	3.208	Today, 13:39:40

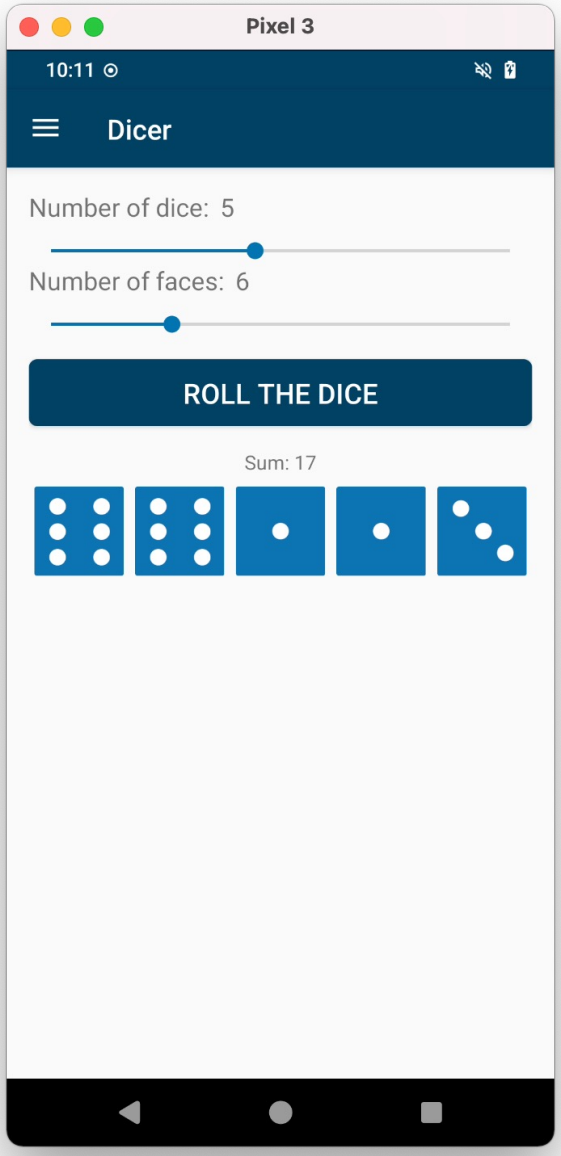
Before

After

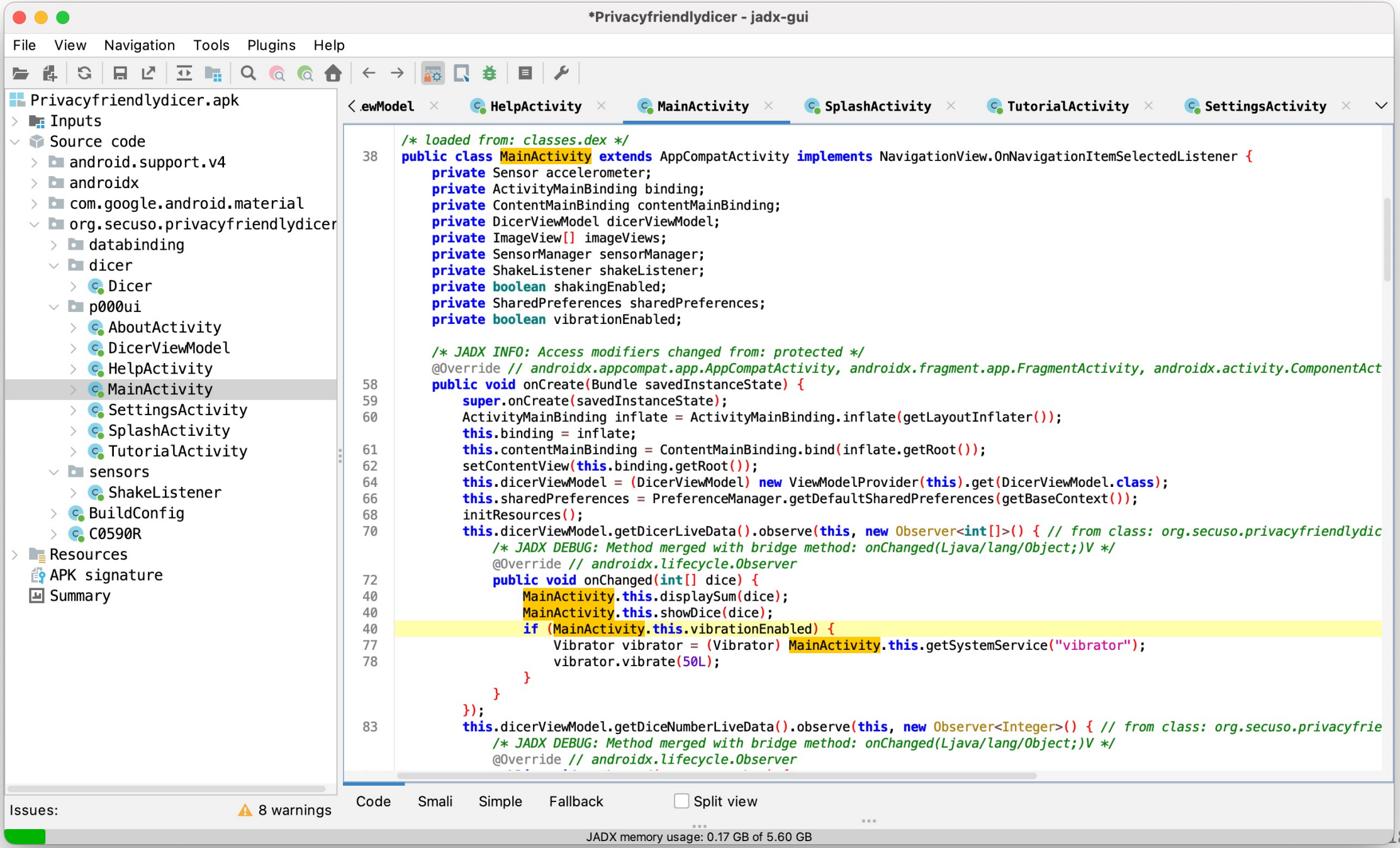
Java Files: 4293 → 276 (6%)

THE DICER

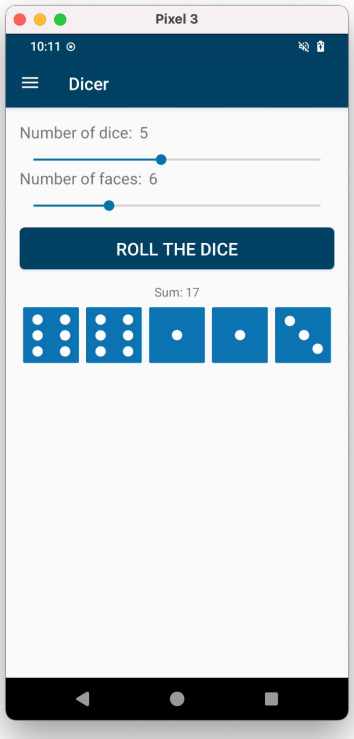




https://f-droid.org/en/packages/org.secuso.privacyfriendlydicer/

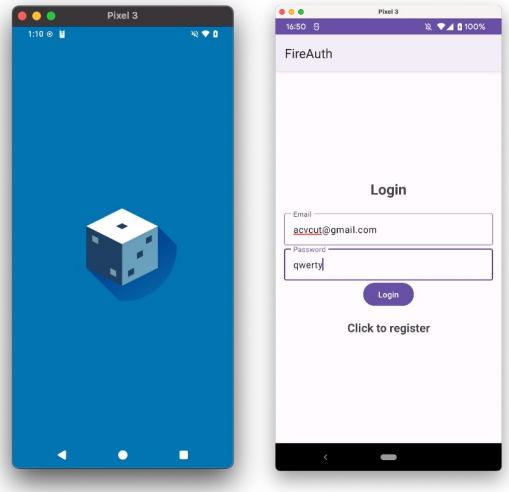

*Privacyfriendlydicer - jadx-gui
File View Navigation Tools Plugins Help
Privacyfriendlydicer.apk
Inputs
Source code
android.support.v4
androidx
com.google.android.material
org.secuso.privacyfriendlydicer
databinding
dicer
Dicer
p000ui
AboutActivity
DicerViewModel
HelpActivity
MainActivity
SettingsActivity
SplashActivity
TutorialActivity
sensors
ShakeListener
BuildConfig
C0590R
Resources
APK signature
Summary
Issues:
8 warnings
ewModel
HelpActivity
MainActivity
SplashActivity
TutorialActivity
SettingsActivity
/* loaded from: classes.dex */
38 public class MainActivity extends AppCompatActivity implements NavigationView.OnNavigationItemSelectedListener {
private Sensor accelerometer;
private ActivityMainBinding binding;
private ContentMainBinding contentMainBinding;
private DicerViewModel dicerViewModel;
private ImageView[] imageViews;
private SensorManager sensorManager;
private ShakeListener shakeListener;
private boolean shakingEnabled;
private SharedPreferences sharedPreferences;
private boolean vibrationEnabled;
/* JADX INFO: Access modifiers changed from: protected */
@Override // androidx.appcompat.app.AppCompatActivity, androidx.fragment.app.FragmentActivity, androidx.activity.ComponentAct
58 public void onCreate(Bundle savedInstanceState) {
59 super.onCreate(savedInstanceState);
60 ActivityMainBinding inflate = ActivityMainBinding.inflate(getLayoutInflater());
this.binding = inflate;
61 this.contentMainBinding = ContentMainBinding.bind(inflate.getRoot());
62 setContentView(this.binding.getRoot());
64 this.dicerViewModel = (DicerViewModel) new ViewModelProvider(this).get(DicerViewModel.class);
66 this.sharedPreferences = PreferenceManager.getDefaultSharedPreferences(getBaseContext());
68 initResources();
70 this.dicerViewModel.getDicerLiveData().observe(this, new Observer<int[]>() { // from class: org.secuso.privacyfriendlydic
/* JADX DEBUG: Method merged with bridge method: onChanged(Ljava/lang/Object;)V */
@Override // androidx.lifecycle.Observer
72 public void onChanged(int[] dice) {
40 MainActivity.this.displaySum(dice);
40 MainActivity.this.showDice(dice);
40 if (MainActivity.this.vibrationEnabled) {
77 Vibrator vibrator = (Vibrator) MainActivity.this.getSystemService("vibrator");
78 vibrator.vibrate(50L);
}
}
});
83 this.dicerViewModel.getDiceNumberLiveData().observe(this, new Observer<Integer>() { // from class: org.secuso.privacyfrie
/* JADX DEBUG: Method merged with bridge method: onChanged(Ljava/lang/Object;)V */
@Override // androidx.lifecycle.Observer
Code
Smali
Simple
Fallback
Split view
JADX memory usage: 0.17 GB of 5.60 GB

```
        .param p1, "this$0"    # Lorg/secuso/privacyfriendlydicer/ui/MainActivity;

        iput-object p1, p0, Lorg/secuso/privacyfriendlydicer/ui/MainActivity$1;->this$0:Lorg/secuso/privacyfriendlydicer/ui/MainActivity;
        invoke-direct {p0}, Ljava/lang/Object;-><init>()V
        return-void
.end method

.method public bridge synthetic onChanged(Ljava/lang/Object;)V
        .locals 0

        check-cast p1, [I
        invoke-virtual {p0, p1}, Lorg/secuso/privacyfriendlydicer/ui/MainActivity$1;->onChanged([I)V
        return-void
.end method

.method public onChanged([I)V
        .locals 3
        .param p1, "dice"    # [I

        iget-object v0, p0, Lorg/secuso/privacyfriendlydicer/ui/MainActivity$1;->this$0:Lorg/secuso/privacyfriendlydicer/ui/MainActivity;
        # invokes: Lorg/secuso/privacyfriendlydicer/ui/MainActivity;->displaySum([I)V
        invoke-static {v0, p1}, Lorg/secuso/privacyfriendlydicer/ui/MainActivity;->access$000(Lorg/secuso/privacyfriendlydicer/ui/MainActivity;[I)V
        iget-object v0, p0, Lorg/secuso/privacyfriendlydicer/ui/MainActivity$1;->this$0:Lorg/secuso/privacyfriendlydicer/ui/MainActivity;
        # invokes: Lorg/secuso/privacyfriendlydicer/ui/MainActivity;->showDice([I)V
        invoke-static {v0, p1}, Lorg/secuso/privacyfriendlydicer/ui/MainActivity;->access$100(Lorg/secuso/privacyfriendlydicer/ui/MainActivity;[I)V
        iget-object v0, p0, Lorg/secuso/privacyfriendlydicer/ui/MainActivity$1;->this$0:Lorg/secuso/privacyfriendlydicer/ui/MainActivity;
        # getter for: Lorg/secuso/privacyfriendlydicer/ui/MainActivity;->vibrationEnabled:Z
        invoke-static {v0}, Lorg/secuso/privacyfriendlydicer/ui/MainActivity;->access$200(Lorg/secuso/privacyfriendlydicer/ui/MainActivity;)Z
        move-result v0
        if-eqz v0, :cond_21
        iget-object v0, p0, Lorg/secuso/privacyfriendlydicer/ui/MainActivity$1;->this$0:Lorg/secuso/privacyfriendlydicer/ui/MainActivity;
        const-string v1, "vibrator"
        invoke-virtual {v0, v1}, Lorg/secuso/privacyfriendlydicer/ui/MainActivity;->getSystemService(Ljava/lang/String;)Ljava/lang/Object;
        move-result-object v0
        check-cast v0, Landroid/os/Vibrator;
        const-wide/16 v1, 0x32
        invoke-virtual {v0, v1, v2}, Landroid/os/Vibrator;->vibrate(J)V
        :cond_21
        return-void
.end method
```



Control flow graph for the ‘ROLL THE DICER’ button

Android app



ACV Shrinking

Element	Ratio	Cov.	Missed	Lines	Missed	Methods	Missed	Classes
1		7.56663%	127009	137406	11081	12447	1689	2026
Total	10397 of 137406	7.56663%	127009	137406	11081	12447	1689	2026

1

1

Element	Ratio	Cov.	Missed	Lines	Missed	Methods	Missed	Classes
1		97.52811%	244	9871	11081	12447	1689	2026
Total	9627 of 9871	97.52811%	244	9871	11081	12447	1689	2026

15

ACVTool 2024

https://github.com/pilgun/acvtool

Fork 28 Starred 89

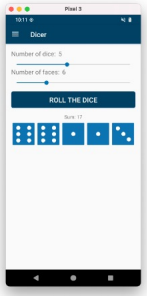
ACVTool 2.3.2
- Implements **MultiDex**
- replaces Apktool with **acvpatcher**
- feature-selected measurement
- experimental **shrinking**

ACVPatcher

> acvpatcher --apkpath base.apk --class ./classes.dex ./classes2.dex

https://github.com/pilgun/acvpatcher

11



21