Apkpatcher

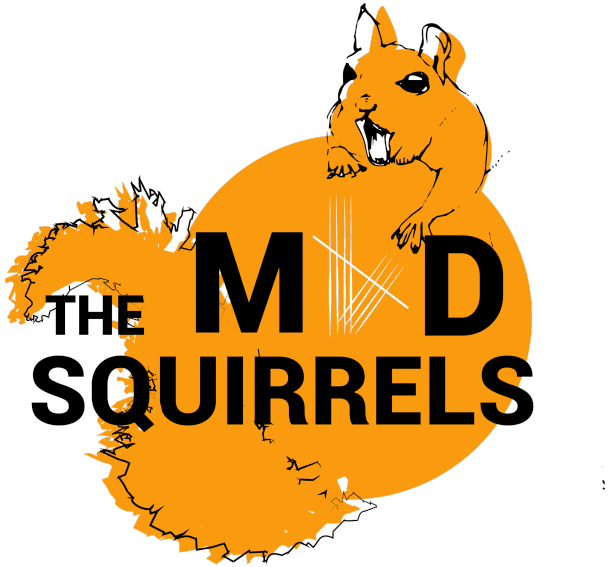
Fast analysis & editing Android Apps without root.

Benoît FORGETTE (MadSquirrels)

05/06/2025

Quarkslab

Who am I ? Contributions ?



Benoît FORGETTE
Software and hardware Security Researcher
topic (Hardware/Android)
SSTIC specialities:
/ Side project presentation

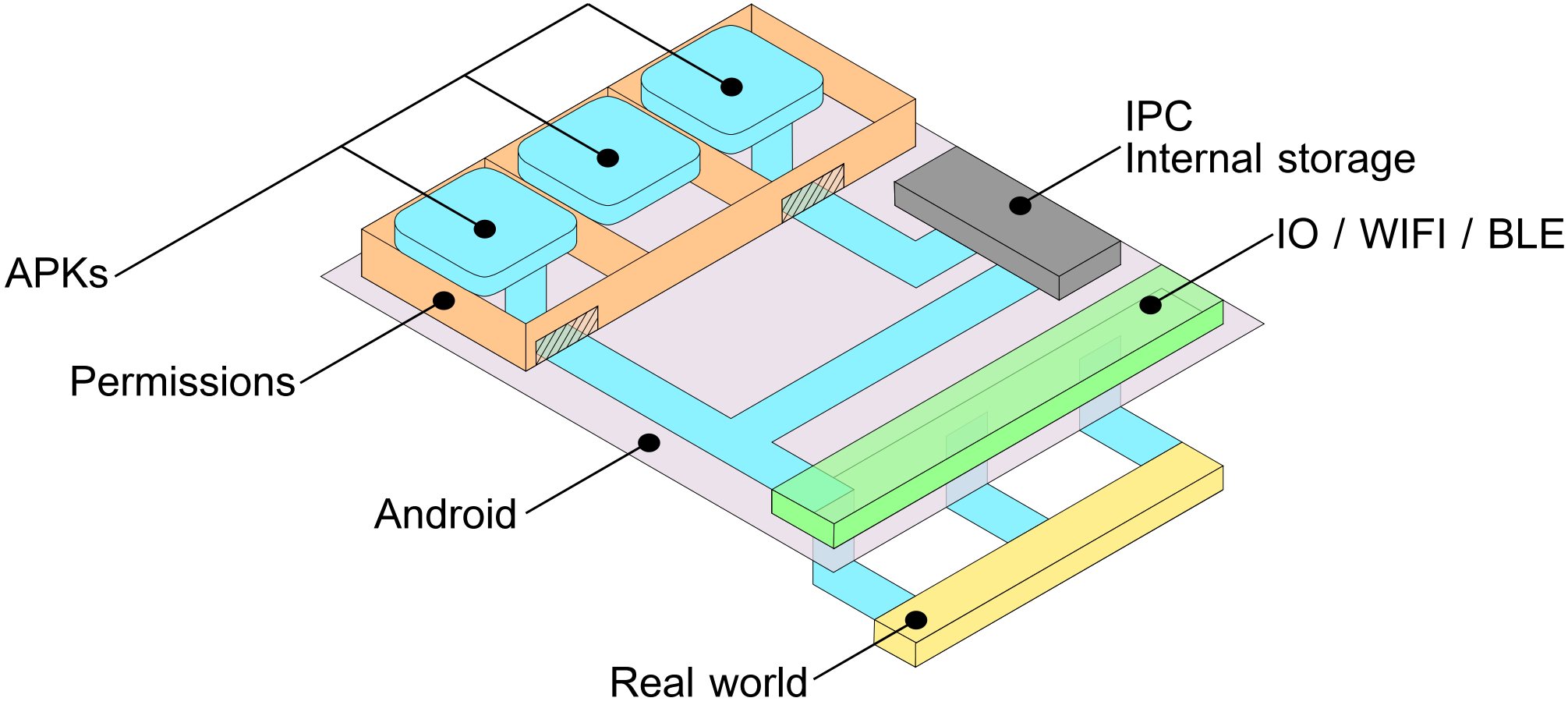
Contributions

- A 4 years tool experience on APK patching with documentation;
- A pythonic tool to manipulate APK;
- A benchmark of patching tool.

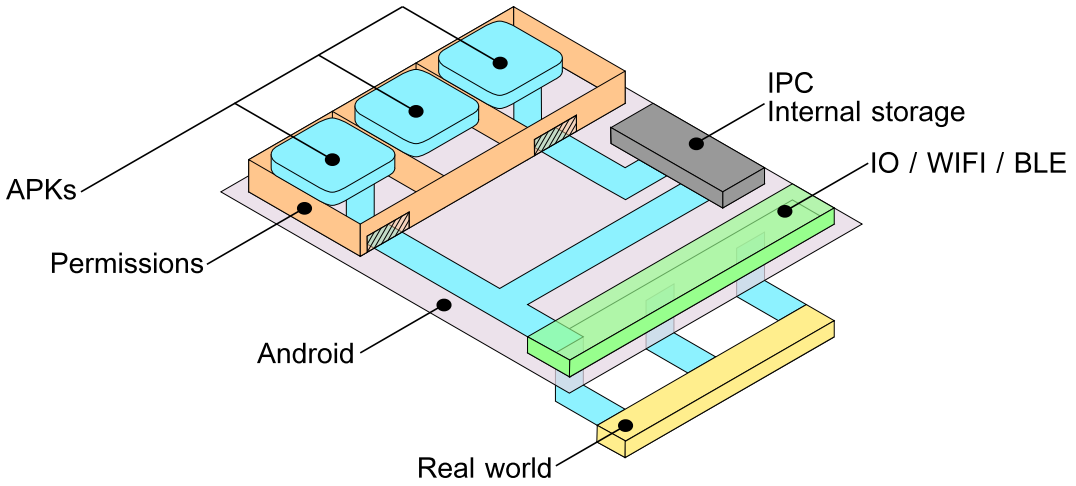
Let's go back to basics.

Quarkslab

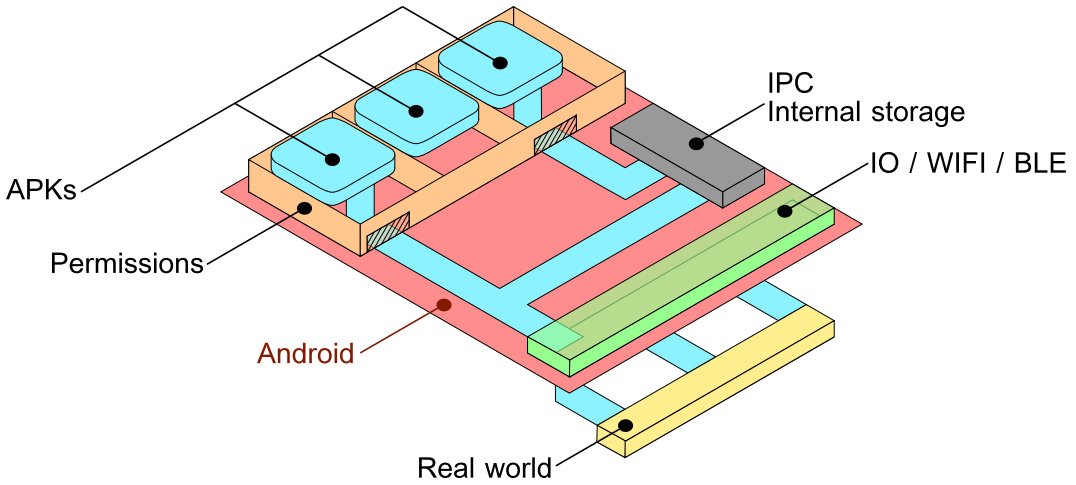
Why we would like to patch



Root an Android

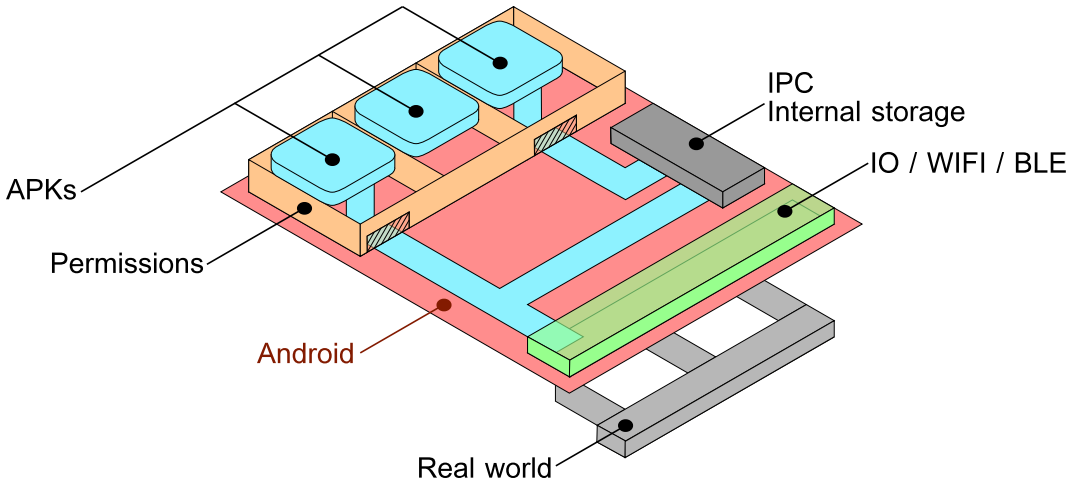


Root an Android



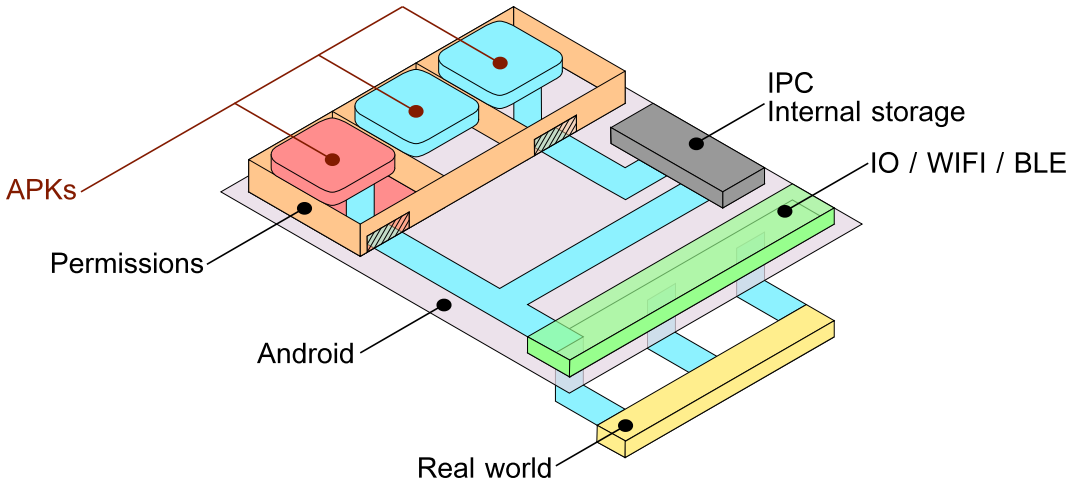
- ❌ A full chain exploitation is needed;
- ❌ It is more commonly know and integrated on detection mechanism;
- ❌ Could not be able to deliver a new APK with modified behavior;
- ✅ Have a full access on system.

Emulate Android



- ❌ Is more commonly know and integrated on detection mechanism;
- ❌ Could not be able to deliver a new APK with modified behavior;
- ❌ Do not have not a full access to external feature;
- ❌ Limited to OS architecture (X86);
- ✅ Have a full access on system.

Patched an APK

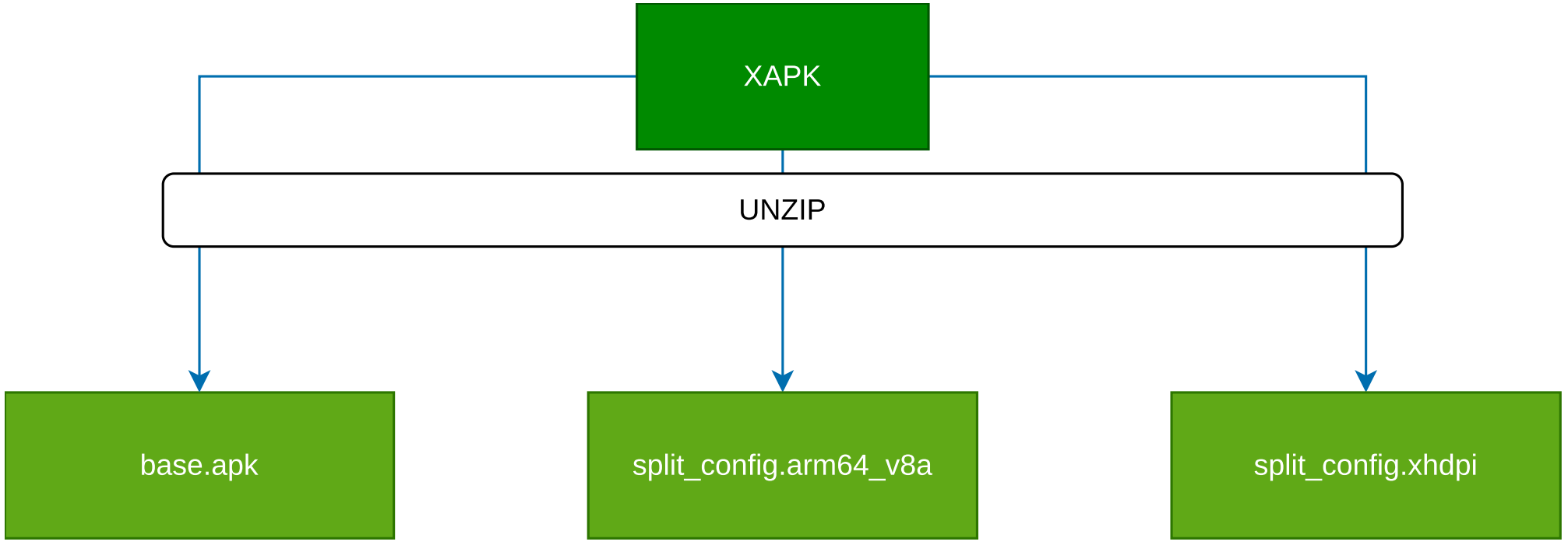


- ❌ Could not by-pass Package manager if a vendor certificate is setup;
- ❌ Need to have an ADB access on Android target;
- ✅ Enable to modify the apk;
- ✅ Can works on any Android system.

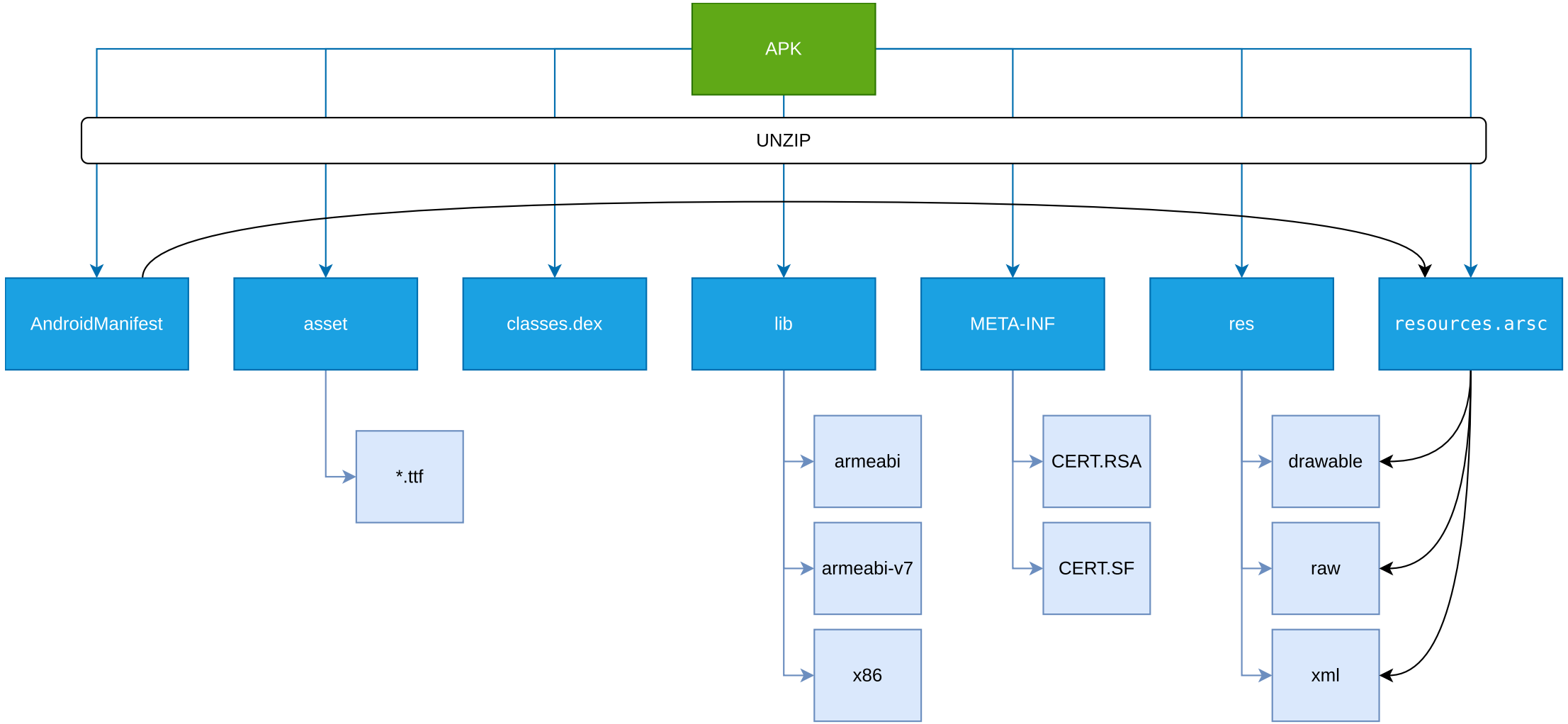
Apkpatcher objective

- A full featured APK repackaging framework;
- A reliable tool;
- Restrict external execution of binary;
- Restrict code to be kept to a minimum based only on the official code released by Android in the sdktools;
- Offer more than a simple tool to inject frida library;
- If possible, do it quickly.

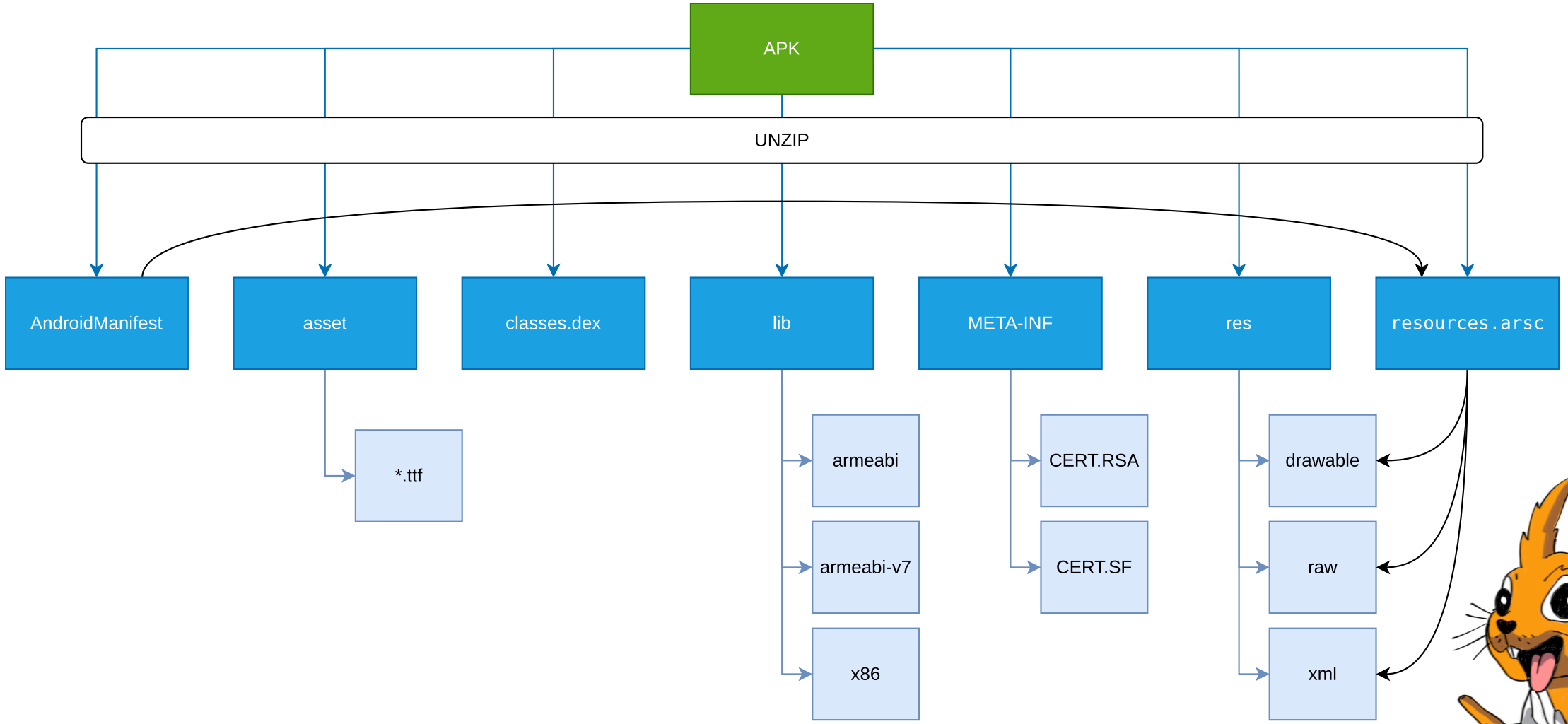
Different format

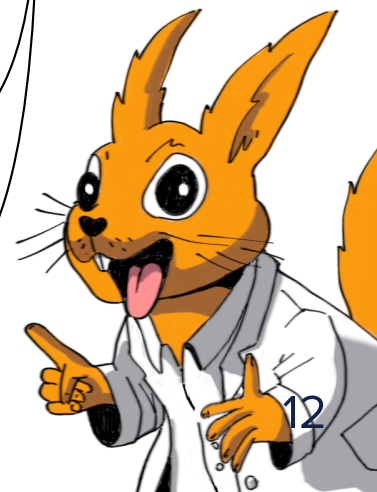


Application structure



Application structure





Application structure

AndroidManifest

```
<manifest package="com.example.pro" platformBuildVersionCode="34" platformBuildVersionName="14">
  <uses-permission android:name="android.permission.ACTIVITY_RECOGNITION"/>
  <application android:theme="@7F16059E" android:label="@7F150398" android:icon="@7F110000" android:debuggable="true"
    android:name="com.fitnesskeeper.runkeeper.RunKeeperApplication"
    android:networkSecurityConfig="@7F18000C" >
    <meta-data android:name="com.google.firebase.messaging.default_notification_icon" android:resource="@7F080413"/>
    <meta-data android:name="com.google.firebase.messaging.default_notification_color" android:resource="@7F06035C"/>
    <activity android:theme="@7F16052C" android:name="com.fitnesskeeper.runkeeper.SplashActivity" android:exported="true">
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>
  </application>
</manifest>
```

Enable debug mode and signed APK in Debug mode

Application structure

AndroidManifest

```
<manifest package="com.example.pro" platformBuildVersionCode="34" platformBuildVersionName="14">
  <uses-permission android:name="android.permission.ACTIVITY_RECOGNITION"/>
  <application android:theme="@7F16059E" android:label="@7F150398" android:icon="@7F110000" android:debuggable="true"
    android:name="com.fitnesskeeper.runkeeper.RunkeeperApplication"
    android:networkSecurityConfig="@7F18000C" >
    <meta-data android:name="com.google.firebase.messaging.default_notification_icon" android:resource="@7F080413"/>
    <meta-data android:name="com.google.firebase.messaging.default_notification_color" android:resource="@7F06035C"/>
    <activity android:theme="@7F16052C" android:name="com.fitnesskeeper.runkeeper.SplashActivity" android:exported="true">
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>
  </application>
</manifest>
```

resources.arsc

```
<public type="mipmap" name="ic_launcher" id="0x7f110000" data="res/9w.png" data_size=8/>
<public type="mipmap" name="ic_launcher" id="0x7f110000" data="res/yn.png" data_size=8/>
<public type="mipmap" name="ic_launcher" id="0x7f110000" data="res/FS.png" data_size=8/>
<public type="mipmap" name="ic_launcher" id="0x7f110000" data="res/RJ.png" data_size=8/>
<public type="mipmap" name="ic_launcher" id="0x7f110000" data="res/o-.png" data_size=8/>
<public type="mipmap" name="ic_launcher" id="0x7f110000" data="res/BW.xml" data_size=8/>
<public type="string" name="global_celsius" id="0x7f150398" data="C" data_size=8/>
<public type="style" name="TripSummary.Card.NoMapNoPhoto.Label" id="0x7f16059e" data="0x0" data_size=0/>
<public type="xml" name="network_security_config" id="0x7f18000c" data="res/8G.xml" data_size=8/>
```

* In the case of a rooted phone in general it is possible to add a certificate to the list of trusted Android certificates.

Application structure

AndroidManifest

```
<manifest package="com.example.pro" platformBuildVersionCode="34" platformBuildVersionName="14">
  <uses-permission android:name="android.permission.ACTIVITY_RECOGNITION"/>
  <application android:theme="@7F16059E" android:label="@7F150398" android:icon="@7F110000" android:debuggable="true"
    android:name="com.fitnesskeeper.runkeeper.RunkeeperApplication"
    android:networkSecurityConfig="@7F18000C" >
    <meta-data android:name="com.google.firebase.messaging.default_notification_icon" android:resource="@7F080413"/>
    <meta-data android:name="com.google.firebase.messaging.default_notification_color" android:resource="@7F06035C"/>
    <activity android:theme="@7F16052C" android:name="com.fitnesskeeper.runkeeper.SplashActivity" android:exported="true">
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>
  </application>
</manifest>
```

resources.arsc

```
<public type="mipmap" name="ic_launcher" id="0x7f110000" data="res/9w.png" data_size=8/>
<public type="mipmap" name="ic_launcher" id="0x7f110000" data="res/yn.png" data_size=8/>
<public type="mipmap" name="ic_launcher" id="0x7f110000" data="res/FS.png" data_size=8/>
<public type="mipmap" name="ic_launcher" id="0x7f110000" data="res/RJ.png" data_size=8/>
<public type="mipmap" name="ic_launcher" id="0x7f110000" data="res/o-.png" data_size=8/>
<public type="mipmap" name="ic_launcher" id="0x7f110000" data="res/BW.xml" data_size=8/>
<public type="string" name="global_celsius" id="0x7f150398" data="C" data_size=8/>
<public type="style" name="TripSummary.Card.NoMapNoPhoto.Label" id="0x7f16059e" data="0x0" data_size=0/>
<public type="xml" name="network_security_config" id="0x7f18000c" data="res/8G.xml" data_size=8/>
```

* In the case of a rooted phone in general it is possible to add a certificate to the list of trusted Android certificates.

Application structure

AndroidManifest

```
<manifest package="com.example.pro" platformBuildVersionCode="34" platformBuildVersionName="14">
  <uses-permission android:name="android.permission.ACTIVITY_RECOGNITION"/>
  <application android:theme="@7F16059E" android:label="@7F150398" android:icon="@7F110000" android:debuggable="true"
    android:name="com.fitnesskeeper.runkeeper.RunkeeperApplication"
    android:networkSecurityConfig="@7F18000C" >
    <meta-data android:name="com.google.firebase.messaging.default_notification_icon" android:resource="@7F080413"/>
    <meta-data android:name="com.google.firebase.messaging.default_notification_color" android:resource="@7F06035C"/>
    <activity android:theme="@7F16052C" android:name="com.fitnesskeeper.runkeeper.SplashActivity" android:exported="true">
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>
  </application>
</manifest>
```

resources.arsc

```
<public type="mipmap" name="ic_launcher" id="0x7f110000" data="res/9w.png" data_size=8/>
<public type="mipmap" name="ic_launcher" id="0x7f110000" data="res/yn.png" data_size=8/>
<public type="mipmap" name="ic_launcher" id="0x7f110000" data="res/FS.png" data_size=8/>
<public type="mipmap" name="ic_launcher" id="0x7f110000" data="res/RJ.png" data_size=8/>
<public type="mipmap" name="ic_launcher" id="0x7f110000" data="res/o-.png" data_size=8/>
<public type="mipmap" name="ic_launcher" id="0x7f110000" data="res/BW.xml" data_size=8/>
<public type="string" name="global_celsius" id="0x7f150398" data="C" data_size=8/>
<public type="style" name="TripSummary.Card.NoMapNoPhoto.Label" id="0x7f16059e" data="0x0" data_size=0/>
<public type="xml" name="network_security_config" id="0x7f18000c" data="res/8G.xml" data_size=8/>
```

8G.xml

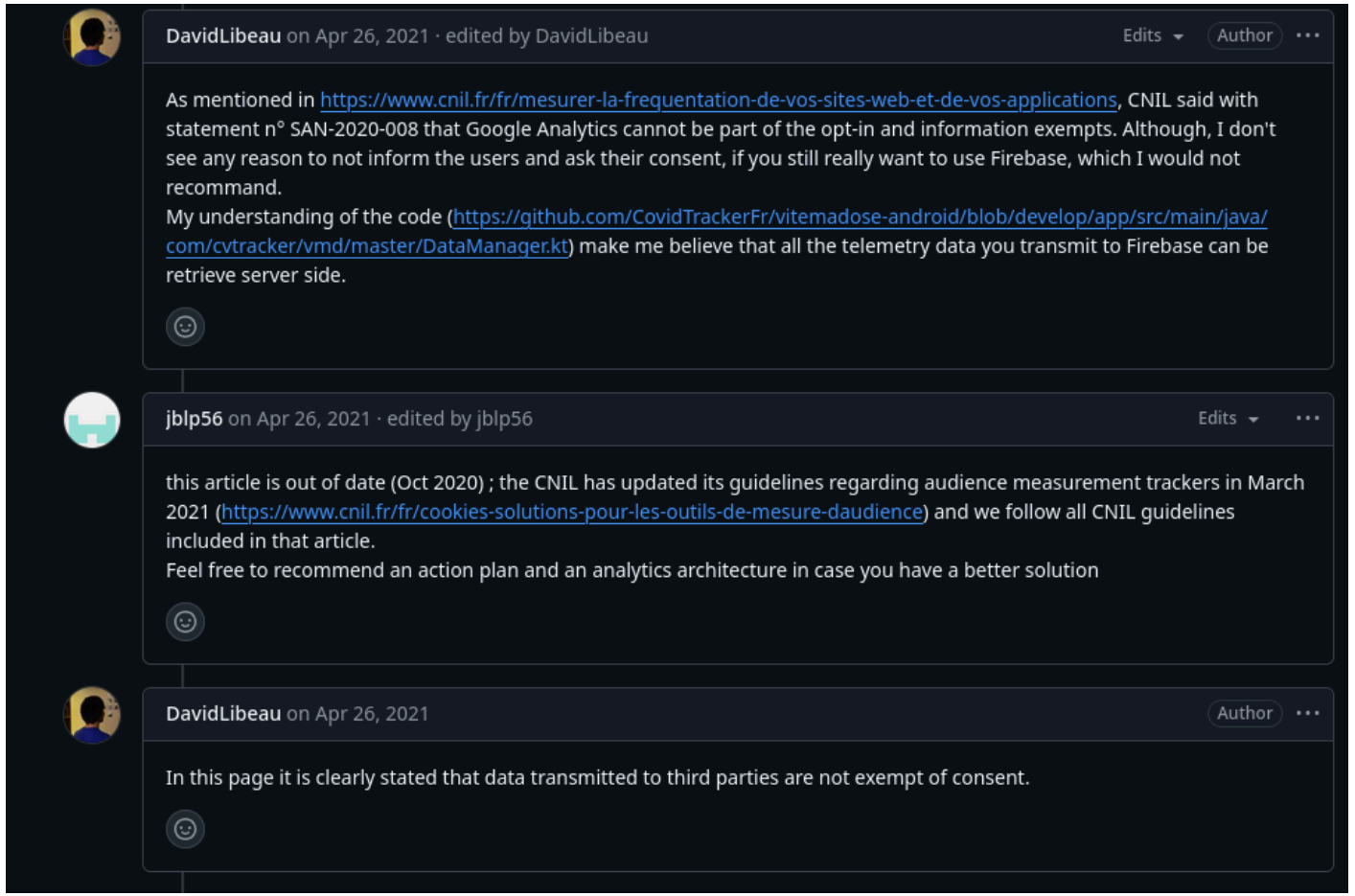
```
<network-security-config>
  <debug-overrides>
    <trust-anchors>
      <certificates src="user"/>
    </trust-anchors>
  </debug-overrides>
</network-security-config>
```

* In the case of a rooted phone in general it is possible to add a certificate to the list of trusted Android certificates.

Let's try Apkpatcher

Quarkslab

Try to remove tracking of ViteMadose



DavidLibeau on Apr 26, 2021 · edited by DavidLibeau — Edits — Author

As mentioned in https://www.cnil.fr/fr/mesurer-la-frequentation-de-vos-sites-web-et-de-vos-applications, CNIL said with statement n° SAN-2020-008 that Google Analytics cannot be part of the opt-in and information exempts. Although, I don't see any reason to not inform the users and ask their consent, if you still really want to use Firebase, which I would not recommand.
My understanding of the code (https://github.com/CovidTrackerFr/vitemadose-android/blob/develop/app/src/main/java/com/cvtracker/vmd/master/DataManager.kt) make me believe that all the telemetry data you transmit to Firebase can be retrieve server side.

jblp56 on Apr 26, 2021 · edited by jblp56 — Edits

this article is out of date (Oct 2020) ; the CNIL has updated its guidelines regarding audience measurement trackers in March 2021 (https://www.cnil.fr/fr/cookies-solutions-pour-les-outils-de-mesure-daudience) and we follow all CNIL guidelines included in that article.
Feel free to recommend an action plan and an analytics architecture in case you have a better solution

DavidLibeau on Apr 26, 2021 — Author

In this page it is clearly stated that data transmitted to third parties are not exempt of consent.

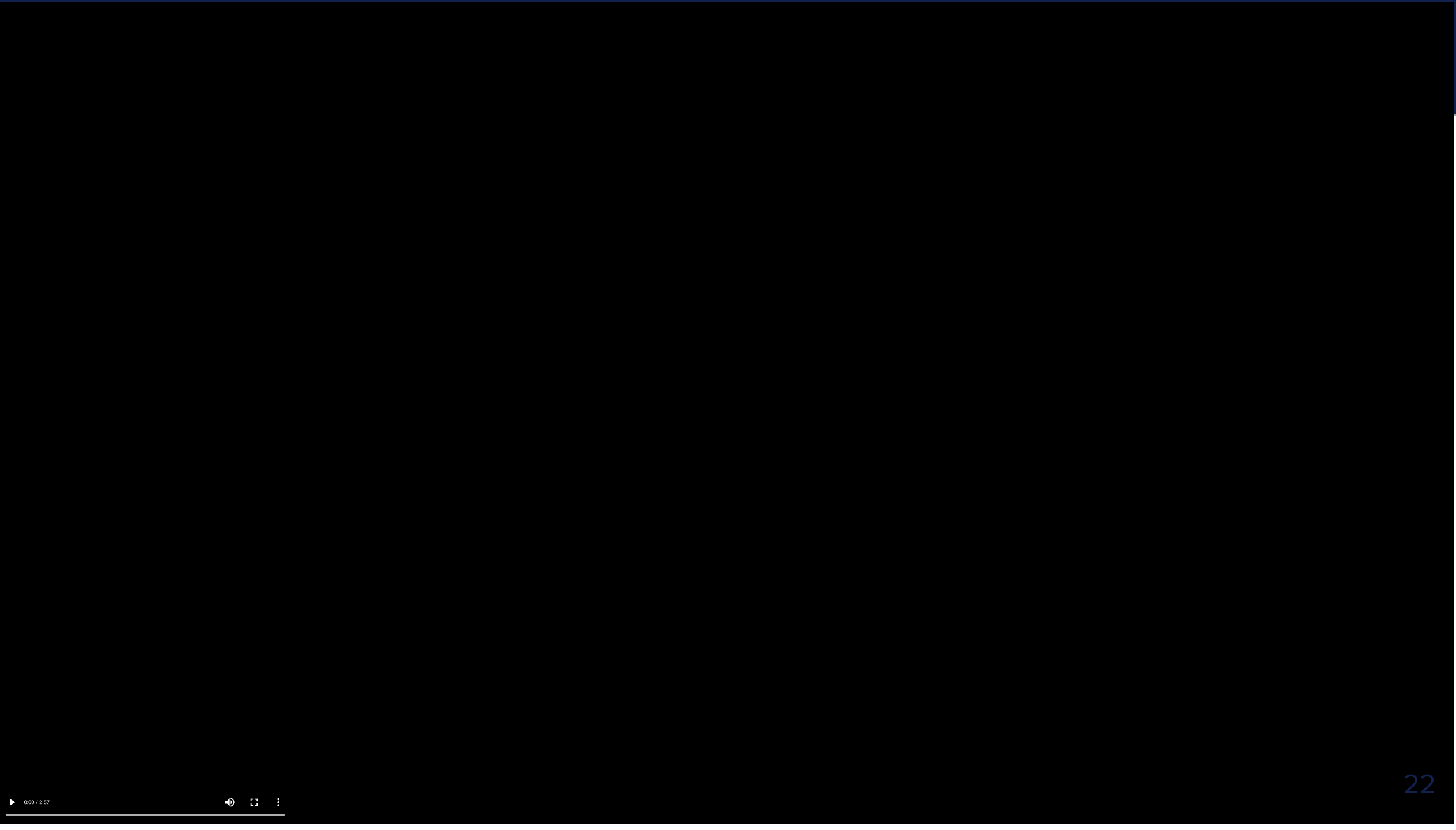
Try to remove tracking of ViteMadose

```
#!/usr/bin/env python

from apkpatcher.smalipatching import Method, get_smali_file_from_class, replace_methods
import click


def replace_method_wrapper(input_dir, classname, prototype, patch):
    m = Method(prototype, patch)
    fname = get_smali_file_from_class(input_dir, classname)
    with open(f"{fname}", "r") as f:
        content = f.read()
    with open(f"{fname}", "w") as f:
        text = replace_methods([m], content)
        content = f.write(text)
```

Try to remove tracking of ViteMadose

```
@click.command()
@click.argument('input_dir', nargs=-1)
def main(input_dir):
    main_dir=input_dir[0]
    prototype = ".method static init(Lcom/google/firebase/FirebaseApp;Lcom/google/firebase/installations/Fi
    patch = """
    .locals 1
    const/4 v0, 0x0
    return-object v0
    """
    classname = "com.google.firebase.crashlytics.FirebaseCrashlytics"
    replace_method_wrapper(input_dir, classname, prototype, patch)


    prototype = ".method public onCreate()Z"
    patch = """
    .locals 1
    const/4 v0, 0x0
    return-object v0
    """
    classname = "com.google.firebase.provider.FirebaseInitProvider"
    replace_method_wrapper(input_dir, classname, prototype, patch)
```

Demo


0:00 / 2:57

Apkpatcher is more than a tool to inject Frida

Main Features of Apkpatcher

- Modify Smali code as native code;
- Library injection, for example Frida to trace an application;
- Proxy certificate injection, for example to add Burp certificate inside the APK;
- Enable debug mode to debug Smali code as Native code.

State of the Art

Tool Name	Version	Date Creation	Project Link
apkpatcher	0.1.25	2021	Apkpatcher Gitlab
apktool	2.11.1	2010	Apktool GitHub
objection	1.11.0	2017	Objection GitHub
apk-patcher	fc517f2	2022	apk-patcher GitHub

State of the Art

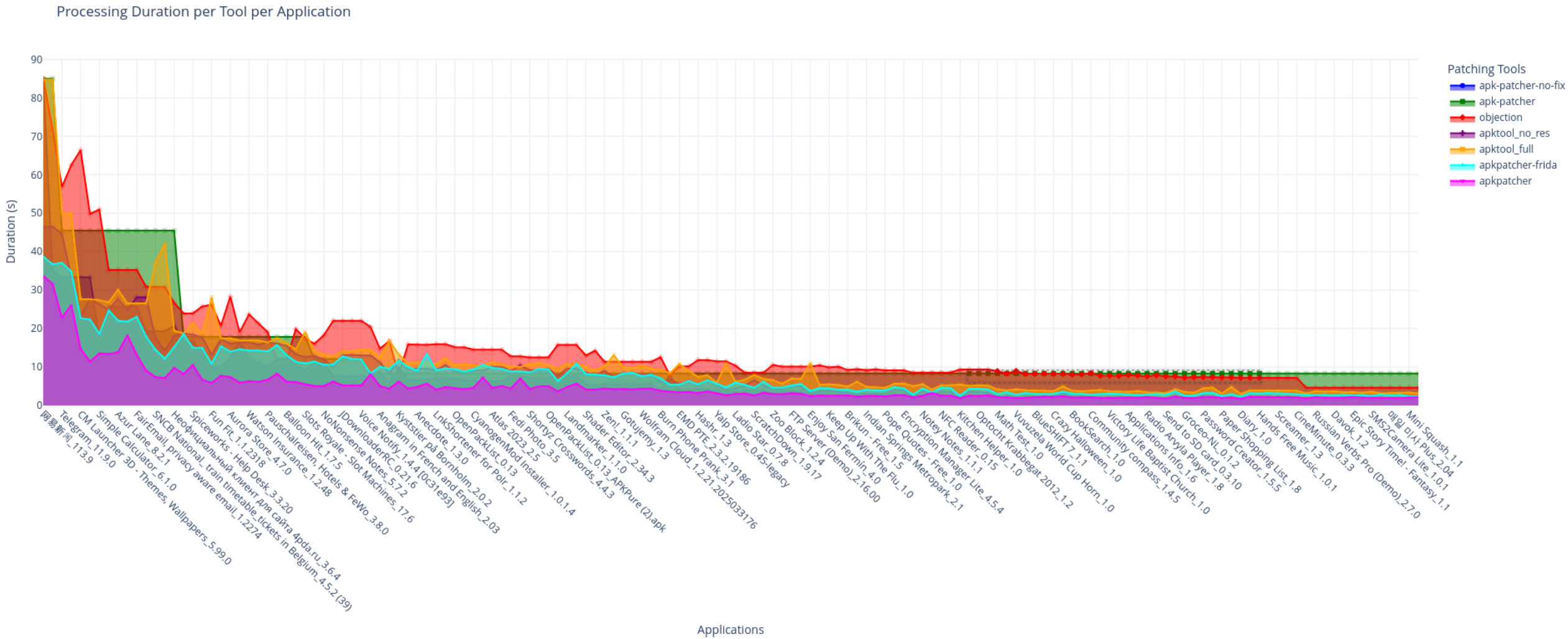
Tool Group	Variant	Unpack & Repack	Inject Frida	With Resources	Allow NativeLib Extraction
apkpatcher	apkpatcher	✅	❌	❌	❌
apkpatcher	apkpatcher-frida	✅	✅	✅	✅
apktool	apktool_no_res	✅	❌	❌	❌
apktool	apktool_full	✅	❌	✅	❌
objection	objection	✅	✅	✅ / ❌	✅
apk-patcher	apk-patcher-no-fix	✅	✅	✅	❌

Benchmark

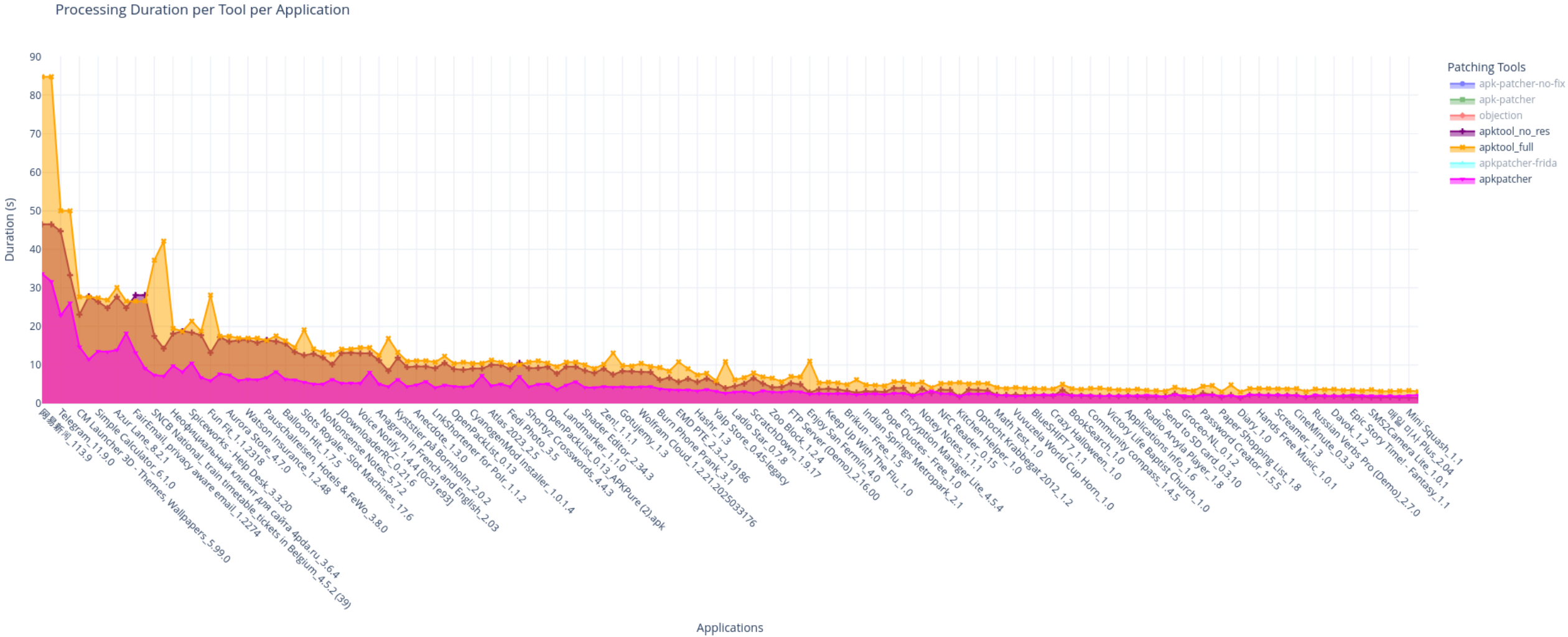
Benchmark carried out on the **kronodroid** dataset on 290 randomly selected applications.

Metric	apkpatcher	apkpatcher-frida	apktool_no_res	apktool_full	objection	apk-patcher-no-fix
Failure Rate (%)	0.0	1.4	4.1	8.1	43.9	47.3
Avg. Speed (s)	4.90	8.18	8.06	10.06	16.55	7.26

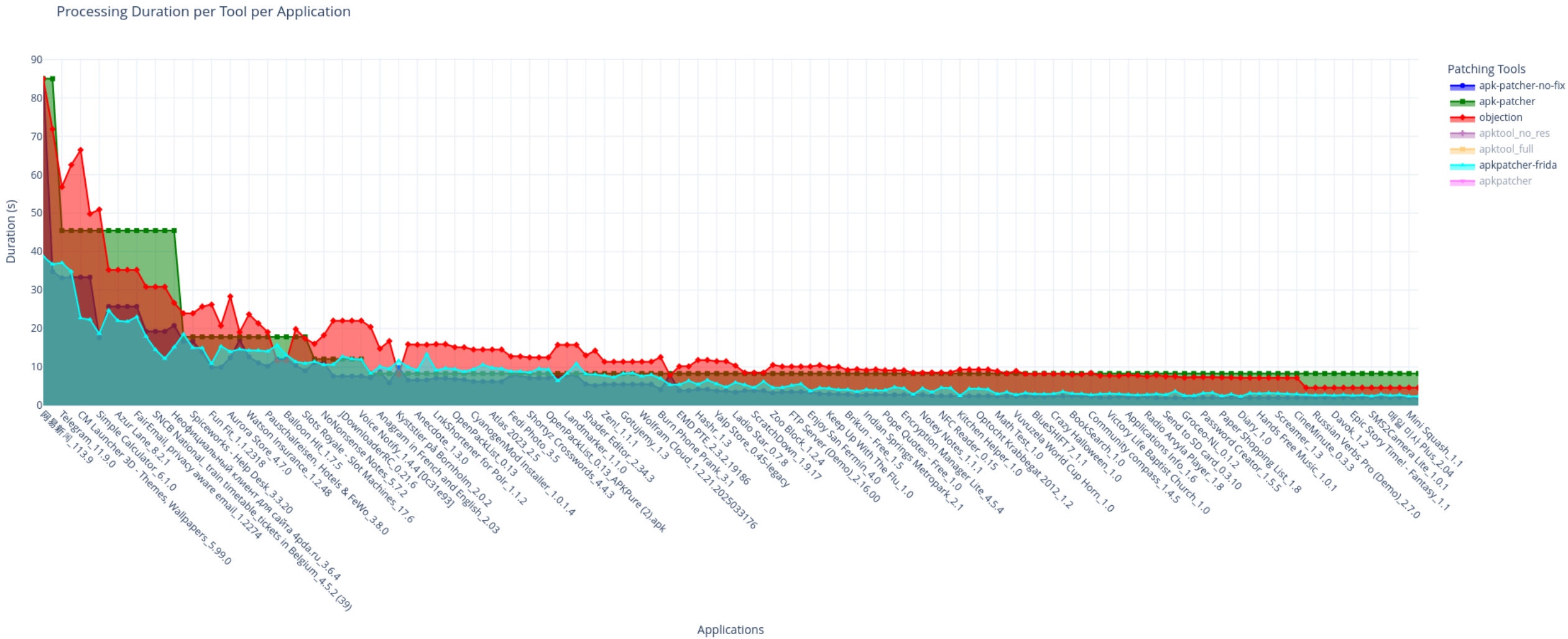
The whole benchmark



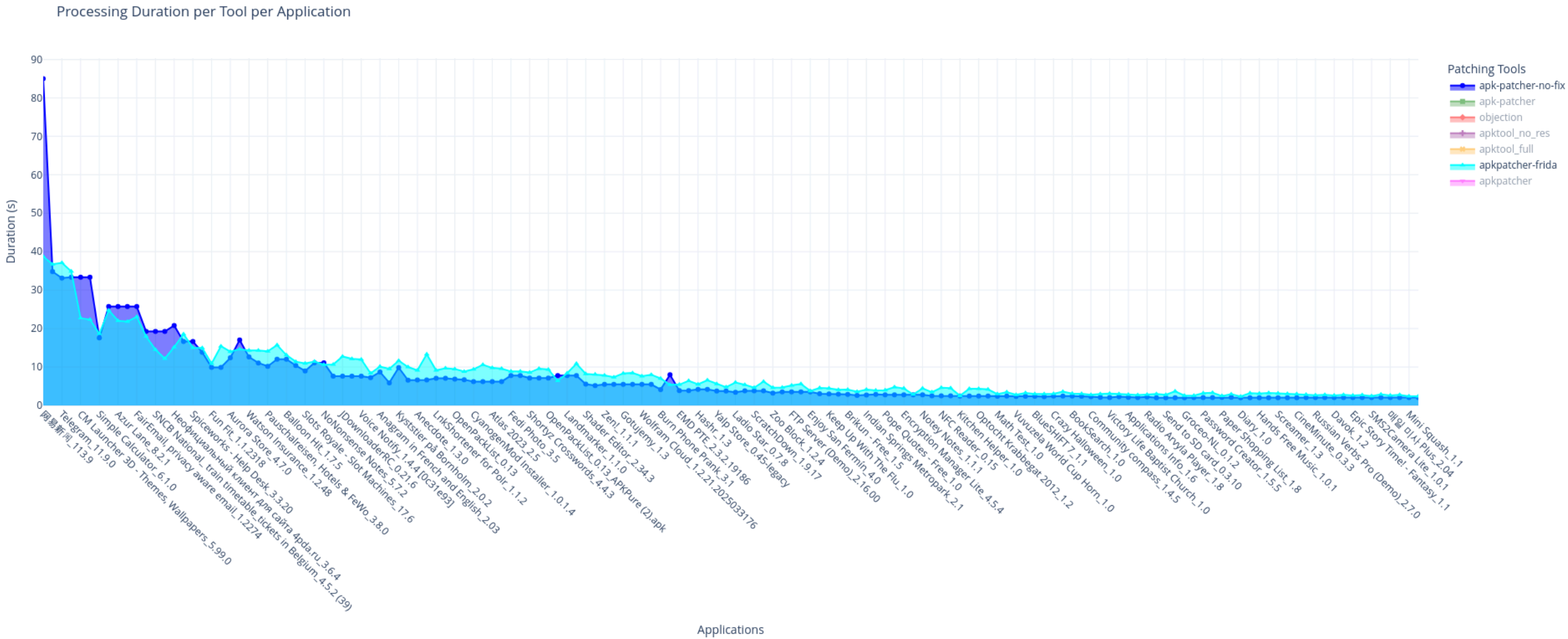
Benchmark of unpack and repacking solution



Benchmark of Frida injection solution



Benchmark of Frida injection solution



Thank you Nodus

Thank you!

Contact information:

apkpatcher: https://apkpatcher.ci-yow.com

Email: bforgette@quarkslab.com

Twitter: https://twitter.com/Mad5quirrel

Quarkslab