目录

安卓逆向调试：XPosed框架

- 最新版本：`v3.0.1`
- 更新时间：`20240729`

简介

介绍用于安卓破解的辅助工具：Xposed框架；先是Xposed框架简介，Xposed发展历史，从Xposed到EdXposed，再到LSPosed；然后介绍背景知识，包括Riru、Zygisk等；再介绍类XPosed框架的VirtualXposed、太极、LSPatch；接着介绍具体如何安装Xposed、EdXposed、LSPosed；其中包括EdXposed常见问题；LSPosed的相关日志、打开LSPosed的app、LSPosed的界面和功能、Shamiko模块；如何具体使用XPosed；以及常见的XPosed的插件模块，包括JustTrustMe，砸壳导出dex的FDex2和dumpdex，app可调试的Xdebuggable和XAppDebug，以及GravityBox；然后介绍如何开发XPosed插件，包括先是用AS新建一个安卓普通app项目，再去改动增加Xposed的配置，写Xposed插件的hook代码，编译和安装Xposed插件即普通安卓apk到安卓手机，然后确认Xposed插件是否安装成功和被正常识别，然后测试Xposed插件是否生效是否能输出log日志；以及贴出Xposed插件的demo项目源代码。最后整理Xposed相关心得；

源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

HonKit源码

- crifan/android_re_xposed_framework: 安卓逆向调试：XPosed框架

如何使用此HonKit源码去生成发布为电子书

详见：crifan/honkit_template: demo how to use crifan honkit template and demo

在线浏览

- 安卓逆向调试：XPosed框架 book.crifan.org
- 安卓逆向调试：XPosed框架 crifan.github.io

离线下载阅读

- 安卓逆向调试：XPosed框架 PDF
- 安卓逆向调试：XPosed框架 ePub
- 安卓逆向调试：XPosed框架 Mobi

版权和用途说明

鸣谢

感谢我的老婆**陈雪**的包容理解和悉心照料，才使得我 `crifan` 有更多精力去专注技术专研和整理归纳出这些电子书和技术教程，特此鸣谢。

其他

作者的其他电子书

本人 `crifan` 还写了其他 `150+` 本电子书教程，感兴趣可移步至：

crifan/crifan_ebook_readme: Crifan的电子书的使用说明

关于作者

关于作者更多介绍，详见：

关于CrifanLi李茂 – 在路上

Xposed简介

- `XPosed`
  - 是什么：本身是一个框架
    - 所以也称：`XPosed框架`
      - = `Xposed Framework`
  - 适用对象：（已root的）安卓手机
  - 作用：可以在XPosed中安装各种插件，实现各种高级功能
    - 引申：常被用来配合破解安卓应用
      - 比如
        - 用 `FDex2` 砸壳导出 `app` 的 `dex` 文件
        - 自己编写 `XPosed` hook插件，实现特定功能
  - 前提：需要安卓手机有`root权限`
  - 特点
    - 开源
    - 支持插件
      - 安装插件后，可以实现各种原本很难实现的效果
  - Github
    - https://github.com/rovo89/XposedBridge
      - rovo89/XposedBridge: The Java part of the Xposed framework.
    - Wiki
      - https://github.com/rovo89/XposedBridge/wiki
        - Using the Xposed Framework API · rovo89/XposedBridge Wiki
          - https://github.com/rovo89/XposedBridge/wiki/Using-the-Xposed-Framework-API
  - 最新的变体
    - `EdXposed`
    - `LSPosed`

Xposed发展历史=各种版本

概述

- 早期 `Android 2.3 ~ 8.1` 的：**Xposed**
- 后来 `Android 8.0 ~ 11` 的：**EdXposed**
- 最新 `Android 8.1 ~ 13` 的：**LSPosed**

详解

- 最早： `Xposed` = `旧版Xposed` = `OG Xposed`
  - 支持Android版本： `Android 2.3` ~ `Android 8.1`
- 后来： `EdXposed`
  - 支持Android版本： `Android 8.0` ~ `Android 11`
    - 已停止维护，不再支持 `安卓11+` 的版本
  - 主页
    - https://edxp.meowcat.org
  - Github
    - EdXposed
      - https://github.com/ElderDrivers/EdXposed
        - A Riru module trying to provide an ART hooking framework (initially for Android Pie) which delivers consistent APIs with the OG Xposed, leveraging YAHFA (or SandHook) hooking framework, supports Android `8.0 ~ 11`
    - EdXposedManager
      - https://github.com/ElderDrivers/EdXposedManager
- 最新： `LSPosed`
  - 支持Android版本： `Android 8.1` ~ `Android 13`
    - 支持最新的 `Android 13`
  - Github
    - https://github.com/LSPosed/LSPosed
      - A Riru / Zygisk module trying to provide an ART hooking framework which delivers consistent APIs with the OG Xposed, leveraging LSPlant hooking framework
  - 官网
    - https://lsposed.org
      - 除了LSPosed的下载地址：
        - https://github.com/LSPosed/LSPosed/releases
      - 之外，啥都没有
        - 或者说：只有一个额外的：（Magisk的MagiskHide替代品的）Shamiko插件 的下载地址
          - https://github.com/LSPosed/LSPosed.github.io/releases

背景知识

Riru

- Riru
  - 概述
    - Riru module providing ART interception framework (natively for Android Pie) that provides consistent API - interfaces with OG Xposed, using YAHFA (or SandHook) interception environment, supports Android 8.0 ~ 11.
  - Github
    - GitHub - RikkaApps/Riru: Inject into zygote process
      - https://github.com/RikkaApps/Riru/
    - 下载
      - Releases · RikkaApps/Riru · GitHub
        - https://github.com/RikkaApps/Riru/releases

Zygisk

什么是Zygisk

- Zygisk is a new feature in Magisk that provides a different way of running root apps and modules
  - Instead of running them as root, Zygisk creates a separate environment for these apps to run in. This environment is called the Zygote process, which is a system process in Android that is responsible for launching and managing app processes.
  - The Zygote process is the very first Android OS process that starts when you boot your phone. This is equivalent to the PID 1 process when you start a Linux based computer or server. Magisk allowing Zygisk to run at the boot level means it can have root access without sending that data to your apps.
- Zygisk是Zygote的Magisk
  - 这将在Zygote进程中运行Magisk的一部分，使Magisk模块更强大。
  - 日常使用中，Zygisk运行在Android核心进程Zygote中。
  - 当一个进程在上述拒绝列表中时，Magisk 将清理该进程的内存空间以确保不应用任何修改，而 root 进程仍然可以在没有 Magisk 监督的情况下通过各种技巧将代码注入其他进程
  - 或者我们更加简单的理解，Zygisk就类似1个新的运行环境，此环境相对于之前的Riru存在天壤之别，这也是为什么zygisk和riru不能同时使用的原因。

Zygisk用途和好处?

- By running root apps and modules in the Zygote process, Zygisk provides several benefits, including improved compatibility, security, and performance. Most importantly, your privacy focused apps like banking apps, Netflix, Google Play Movies etc. will not detect your rooted device and will continue to work like normal.
  - improves compatibility
  - enhances security
  - improve performance
- 引入Zygisk后，传统MagiskHide已经被取消，但是我们可以在面具设置里，打开Zygisk按钮再Zygisk环境下，多出来1个排除列表。
  - 在排除列表进入后，界面非常类似之前的Magisk Hide功能，通过对排除列表里部分软件打钩，成功对软件实现了ROOT隐藏。
  - 除以上这些打开Zygisk后，我们刷入相关模块后，只需要直接刷入功能模块即可实现。
  - 如LSPosed分riru版和Zygisk版本：
    - Riru版本激活LSP需要刷入riru+lsp两个zip才能实现
    - 而Zygisk版本直接刷1个ZyLSP即可快速激活
      - 操作起来更加简单

类Xposed框架

除了Xposed框架本身，后来出现很多**类Xposed框架**，主要是无需root即可体验Xposed插件的效果。

相关类Xposed框架整理如下：

VirtualXposed

- TODO
  - 【未解决】小米9中安装VirtualXposed
  - 【未解决】小米9中无法运行VirtualXposed会崩溃一闪而过
  - 【已解决】VirtualXposed是否真的免Root和支持哪些插件
  - 【已解决】网易MuMu中安装VirtualXposed
  - 【未解决】网易MuMu中VirtualXposed中安装和使用JustTrustMe

- VirtualXposed
  - 一句话简介：Use Xposed with a simple APP, without needing to root, unlock the bootloader, or flash a system image
  - 介绍：一个类似于Xposed框架
    - 但
      - 优点
        - 无需Root
      - 缺点
        - 有些其他的限制?
  - 主页
    - Download | VirtualXposed
      - https://vxposed.com

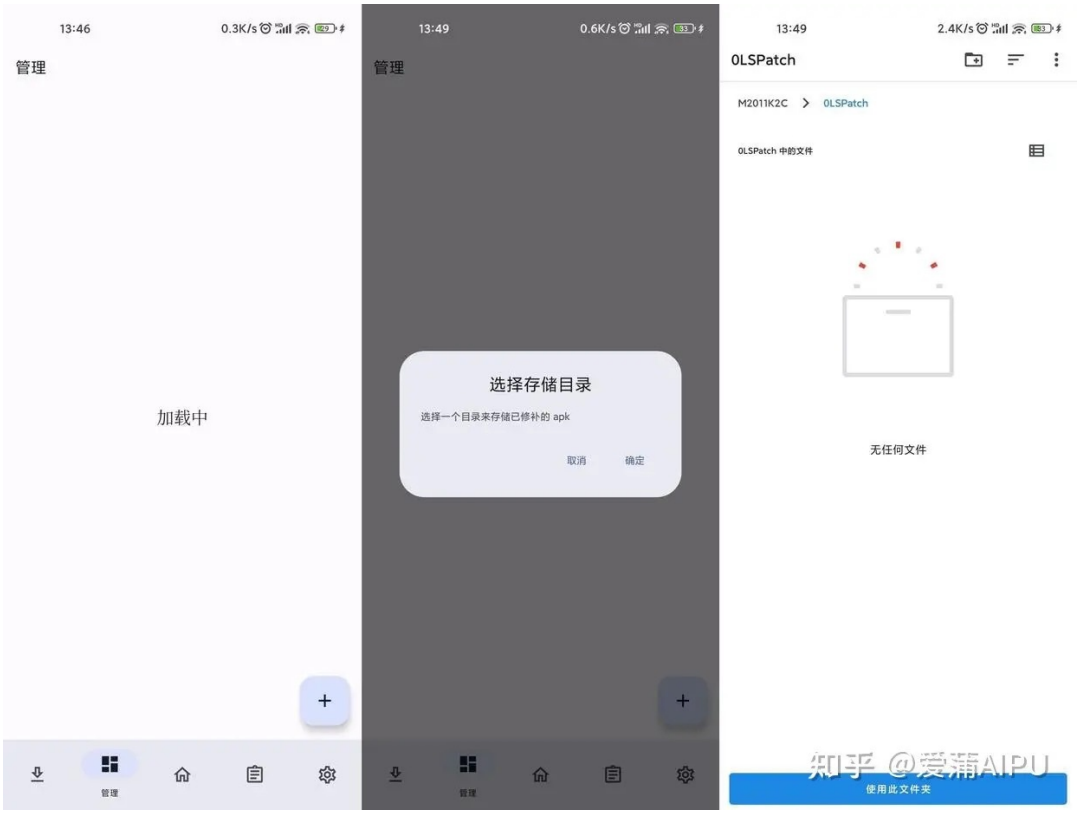
太极= `Tai Chi`

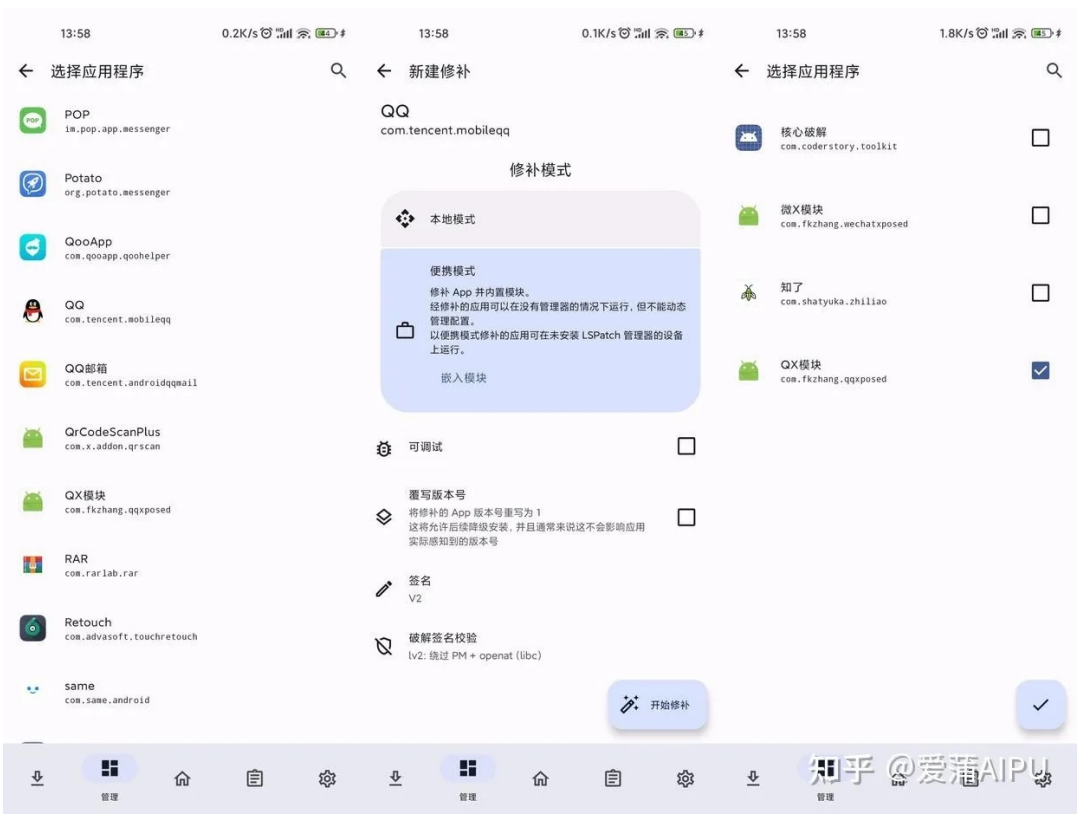
- 太极·虚拟框架
  - 一句话简介：免解锁,免Root,就能使用 `Xposed` 框架
    - 一个可以免 `Root` 运行的类 `Xposed` 框架
  - 资料
    - 官网
      - https://taichi.cool/zh/
    - 文档
      - 介绍 | 太极
        - https://taichi.cool/zh/doc/
    - 下载
      - https://taichi.cool/zh/download.html
    - 模块下载
      - 太极官网-太极app，开启全新的XPOSED模块使用体验
        - https://www.taichi-app.com/#/download/alipay

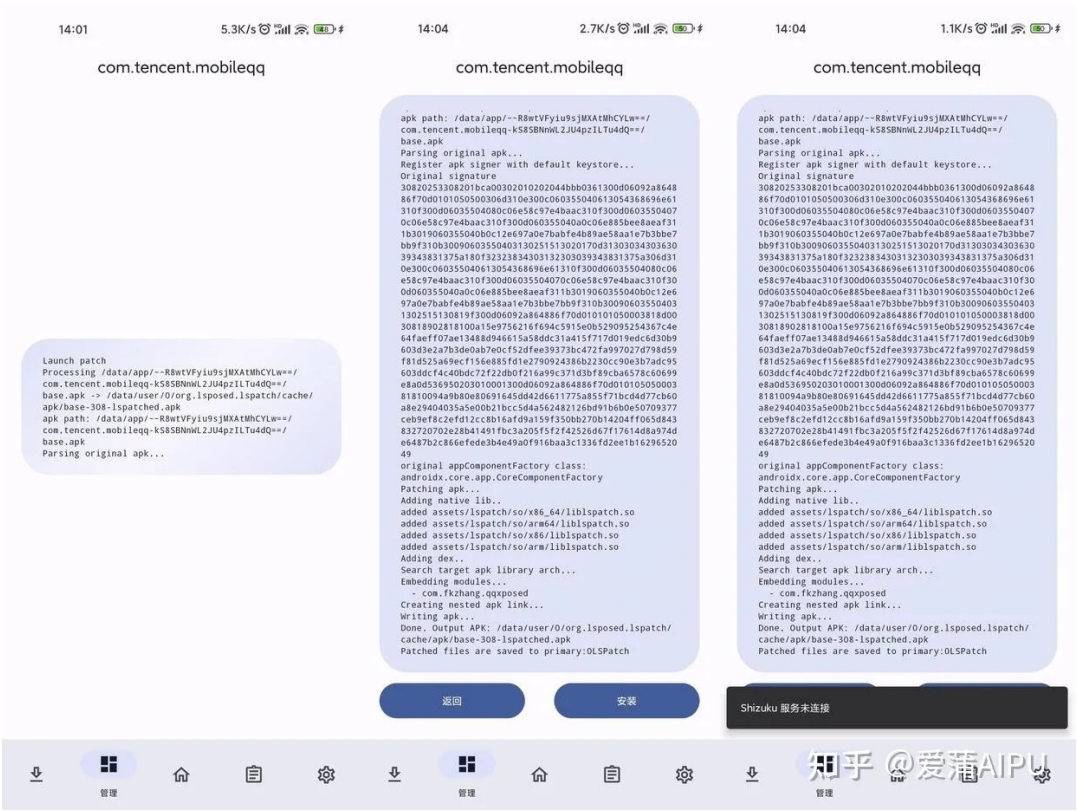
LSPatch

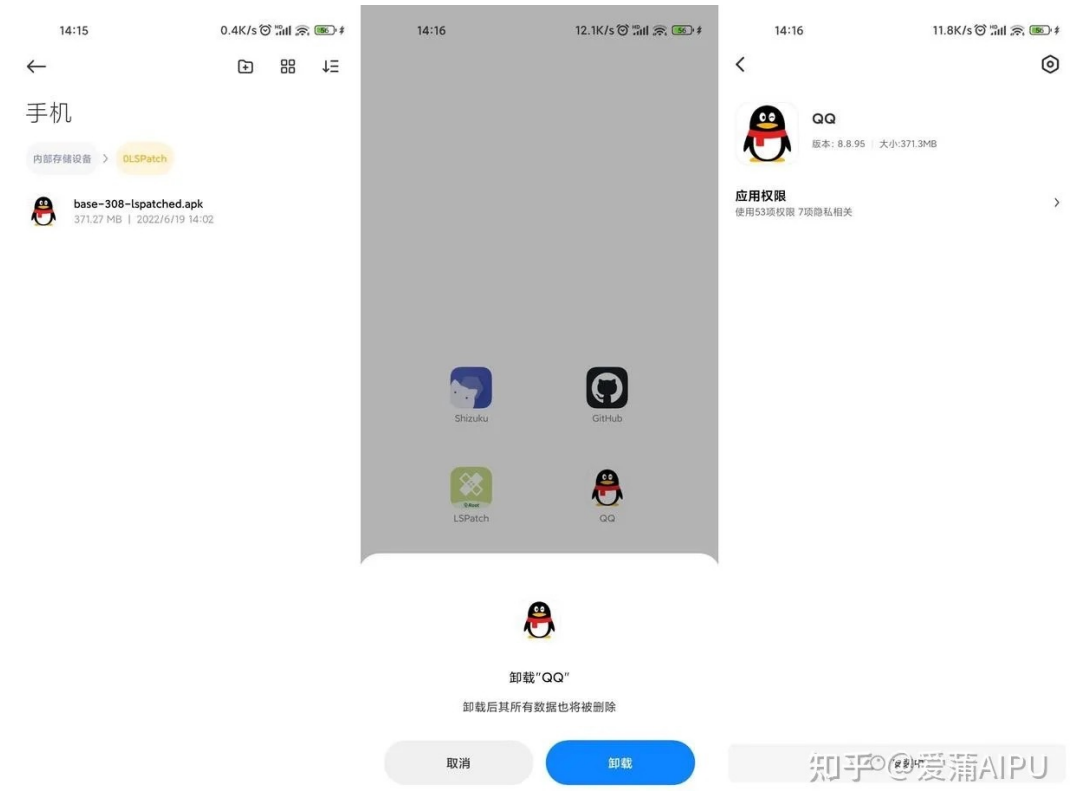
- `LSPatch`
  - Github
    - https://github.com/LSPosed/LSPatch
  - 概述
    - A non-root Xposed framework extending from LSPosed
      - 基于LSPosed的免Root的Xposed框架软件
  - 特定
    - （安卓手机）无需root（无需解锁BL）
  - 用途=作用
    - 给选定的安卓应用，植入Xposed模块，实现各种插件的功能
  - 图
    - LSPatch
      - 

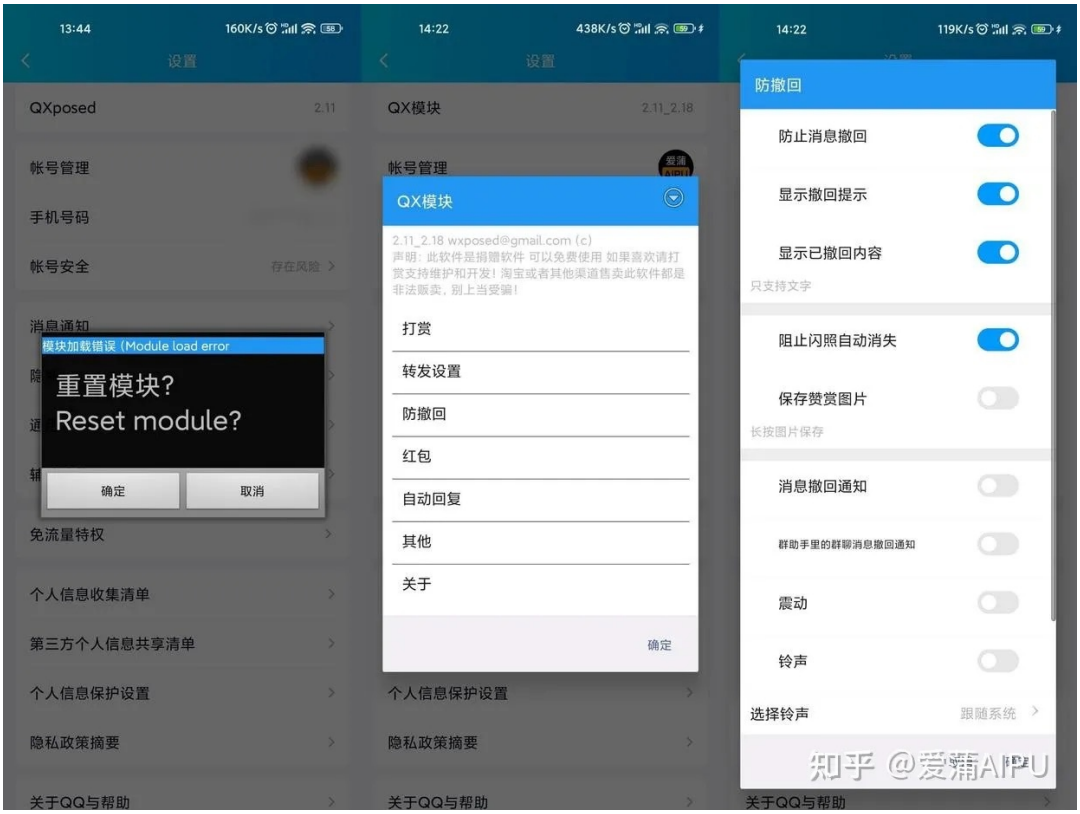
    - 给QQ安装插件


13:46
0.3K/s
管理
加载中
管理
13:49
0.6K/s
管理
选择存储目录
选择一个目录来存储已修补的 apk
取消
确定
管理
13:49
2.4K/s
0LSPatch
M2011K2C
0LSPatch
0LSPatch 中的文件
无任何文件
使用此文件夹
知乎 @爱浦AIPU


13:58
0.2K/s
选择应用程序
POP
im.pop.app.messenger
Potato
org.potato.messenger
QooApp
com.qooapp.qoohelper
QQ
com.tencent.mobileqq
QQ邮箱
com.tencent.androidqqmail
QrCodeScanPlus
com.x.addon.qrscan
QX模块
com.fkzhang.qqxposed
RAR
com.rarlab.rar
Retouch
com.advasoft.touchretouch
same
com.same.android
管理
13:58
0.1K/s
新建修补
QQ
com.tencent.mobileqq
修补模式
本地模式
便携模式
修补 App 并内置模块。
经修补的应用可以在没有管理器的情况下运行，但不能动态管理配置。
以便携模式修补的应用可在未安装 LSPatch 管理器的设备上运行。
嵌入模块
可调试
覆写版本号
将修补的 App 版本号重写为 1
这将允许后续降级安装，并且通常来说这不会影响应用实际感知到的版本号
签名
V2
破解签名校验
lv2: 绕过 PM + openat (libc)
开始修补
管理
13:58
1.8K/s
选择应用程序
核心破解
com.coderstory.toolkit
微X模块
com.fkzhang.wechatxposed
知了
com.shatyuka.zhiliao
QX模块
com.fkzhang.qqxposed
管理
知乎 @爱浦AIPU


14:01
5.3K/s
com.tencent.mobileqq
Launch patch
Processing /data/app/~~R8wtVFyiu9sjMXAtMhCYLw==/com.tencent.mobileqq-kS8SBNnWL2JU4pzILTu4dQ==/base.apk -> /data/user/0/org.lsposed.lspatch/cache/apk/base-308-lspatched.apk
apk path: /data/app/~~R8wtVFyiu9sjMXAtMhCYLw==/com.tencent.mobileqq-kS8SBNnWL2JU4pzILTu4dQ==/base.apk
Parsing original apk...
14:04
2.7K/s
com.tencent.mobileqq
apk path: /data/app/~~R8wtVFyiu9sjMXAtMhCYLw==/com.tencent.mobileqq-kS8SBNnWL2JU4pzILTu4dQ==/base.apk
Parsing original apk...
Register apk signer with default keystore...
Original signature
Patching apk...
Adding native lib..
added assets/lspatch/so/x86_64/liblspatch.so
added assets/lspatch/so/arm64/liblspatch.so
added assets/lspatch/so/x86/liblspatch.so
added assets/lspatch/so/arm/liblspatch.so
Adding dex..
Search target apk library arch...
Embedding modules...
- com.fkzhang.qqxposed
Creating nested apk link...
Writing apk...
Done. Output APK: /data/user/0/org.lsposed.lspatch/cache/apk/base-308-lspatched.apk
Patched files are saved to primary:OLSPatch
返回
安装
14:04
1.1K/s
com.tencent.mobileqq
Shizuku 服务未连接
管理
管理
管理
知乎 @爱蒲AIPU


14:15
0.4K/s
手机
内部存储设备
OLSPatch
base-308-lspatched.apk
371.27 MB | 2022/6/19 14:02
14:16
12.1K/s
Shizuku
GitHub
LSPatch
QQ
卸载"QQ"
卸载后其所有数据也将被删除
取消
卸载
14:16
11.8K/s
QQ
版本: 8.8.95 大小:371.3MB
应用权限
使用53项权限 7项隐私相关
知乎 @爱蒲AIPU

- 

安装Xposed

下面分别介绍，如何安装：

- `Xposed` 的 `Xposed Installer`
- `EdXposed` 的 `EdXposed Manager`
- `LSPosed` （自带Manager的app）

安装XPosed Installer

TODO：

- 【已解决】Mac中夜神安卓模拟器中安装Xposed框架
- 【记录】给二手已root小米4设置Charles代理和安装Charles证书和启用Xposed
- 【已解决】用WrBug的DumpDex从app中hook导出dex文件
- 【未解决】小米4中尝试安装最新3.1.5的Xposed Installer去解决2.6版本提示的不兼容的问题
- 【已解决】mac中试用FDex2去hook导出安卓app的dex等文件
- 【已解决】Android 4.4.4的小米4Xposed Installer出错：Xposed目前不兼容Android SDK版本19或您的处理器架构armeabi-v7a
- 【已解决】Mac中夜神模拟器中安装Xposed框架
- 【已解决】价格便宜但支持root的Android手机
- 【已解决】如何确定此二手小米4是否的确已经root
- 【记录】二手已root的小米4安卓手机照片和信息
- 【已解决】小米4的MIUI系统自动升级导致清楚已有root权限后如何恢复root权限
- 【已解决】小米4中重新安装Xposed Installer和激活Xposed框架

- 说明
  - `旧版本的Xposed` == `Xposed Installer`
- 如何安装 `Xposed Installer`
  - 一般是通过 `Xposed Installer` = `Xposed安装器` 来安装 `Xposed框架`，安装到安卓系统中
    - Xposed Installer
      - 可以理解为是一个**特殊**的安卓的应用
        - 专门用来安装Xposed框架
- 被安装的安卓系统
  - 可以用 `安卓模拟器`
    - 安卓模拟器有很多，不是所有的都能成功安装
      - 尝试过
        - `网易MuMu`
          - 运行速度不错，但不支持设置Wifi代理
            - 详见
              - 【未解决】Mac中用Charles抓包网易Mumu安卓模拟器中Android的app
        - `Andy`
          - 安装后无法正常运行
            - 详见
              - 【未解决】Mac中尝试用Andy安卓模拟器去供Charles抓包Android中app的数据
      - 最终跑通的路是
        - Mac版 `夜神安卓模拟器`
          - 基于 `Android 4.4.2`
            - -》找到了匹配的 `2.7`（或更低的 `2.6` 等）版本的 `Xposed Installer`，即可安装

- 注：更高的 `3.x` 版本，比如 `3.5.1` ，不支持 `Android 4.4` ，只支持 `Android 5`
  - 详见
    - 【已解决】Mac中夜神安卓模拟器中安装Xposed框架
      - 安装XPosed框架 · 好用的安卓模拟器：夜神Nox (crifan.org)
    - 【已解决】Mac中夜神模拟器中安装Xposed模块：JustTrustMe

- 也可以用 `安卓真机`
  - 但前提是：安卓手机要有root权限
    - 但（2019年及之后）很难买到支持root权限的手机
      - 之前跑通的路是
        - 淘宝上买的**二手**的**已root**的 `小米4` ，型号： `MI 4LTE-CU` ，系统： `Android 4.4.4`
          - 注意：
            - 在无端被 `MIUI` 自动升级（从 `5.8.5` 升级为 `7.5.12.17` ）而丢失了卖家预装好的 `Xposed Installer` 之后，再去费了番功夫想办法重新安装可用的 `Xposed` ，最终思路是：
              - 网上找到了某大神 `SolarWarez` 修改后的 `2.6` 的版本的 `Xposed` ：
                - `XposedInstaller_v2.6.1_by_SolarWarez_20151129.apk`

    - 才得以成功安装
    - 激活Xposed后，即可正常安装 `JustTrustMe` 等模块
        - 最终实现 `SSL pinning` 的绕开/禁止的效果
            - 就可以全部看到https解密后的明文了
    - 详见
        - 【已解决】小米4中重新安装Xposed Installer和激活Xposed框架
        - 【记录】给二手已root小米4设置Charles代理和安装Charles证书和启用Xposed

注意手机有变砖可能

刷xposed框架会修改系统文件，所以可能会导致手机变砖，系统崩溃。以及刷成功后系统变卡变慢。并且不保证每台机器都可以刷成功，请自行评估在决定是否安装

-》不建议在自己的（常用）手机中装XPosed，因为很容易导致变砖

-》建议在安卓模拟器上，或者其他开发专用的安卓机上装 `XPosed`

-》或者也可以在自己手机或开发专用安卓机中**免root**安装 `VirtualXposed` 或 `太极`

因为免root不会导致系统崩溃或手机变砖

安装EdXposed

此处介绍 `Android 11` 的 `Google Pixel 3` 中如何安装 `EdXposed`：

要安装 `EdXposed框架` 的话，要：

- 先安装好： `Magisk`
  -


10:11
45%
主页
请确认正在使用开源的 Magisk Manager。未知来源的 Manager 可能具有恶意行为。
不再显示
Magisk
更新
最新 22.1 (22100)
当前 21.0 (21000)
Ramdisk 是
A/B 是
SAR 是
Crypto File
Manager
更新
最新 22.1 (22100) (20)
当前 8.0.0 (302)
包名 com.topjohnwu.magisk
测试 SafetyNet 证明
卸载 Magisk
支持开发
Magisk 将一直保持免费且开源，向开发者捐赠以表示支持。
@topjohn
@diareuse

- 再安装2部分
  - `Magisk` 的插件
    - 先：`Riru` = `Riru Core`
    - 再：`EdXposed`
      - 依赖于：`Riru插件`
      - 是：`EdXposed` 的 `Core` 端
  - `EdXposed Manager` = `EdXposed的app端`

或者换种说法：

- `EdXposed` 分2部分
  - `EdXposed` 的**core**内核
    - Magisk中的：`EdXposed插件` = `Riru-EdXposed`
      - 其依赖于：Magisk插件：`Riru`
  - `EdXposed` 的**app**端
    - `EdXposed Manager` 的apk

所以分别去下载和安装：

- `EdXposed Manager`
  - Releases · ElderDrivers/EdXposedManager (github.com)
    - https://github.com/ElderDrivers/EdXposedManager/releases/download/v4.6.2.0.0/EdXposedManager-4.6.2-46200-org.meowcat.edxposed.manager-release.apk
- `Magisk插件`
  - `Riru (Core)`
    - Releases · RikkaApps/Riru (github.com)
      - https://github.com/RikkaApps/Riru/releases/download/v25.4.4/riru-v25.4.4-debug.zip
  - `EdXposed`
    - Releases · ElderDrivers/EdXposed (github.com)
      - https://github.com/ElderDrivers/EdXposed/releases/download/v0.5.2.2/EdXposed-v0.5.2.2_4683-master-z-debug.zip

安装过程：

- Magisk的插件
  - `Riru插件` = `Riru` = `Riru Core`
    -

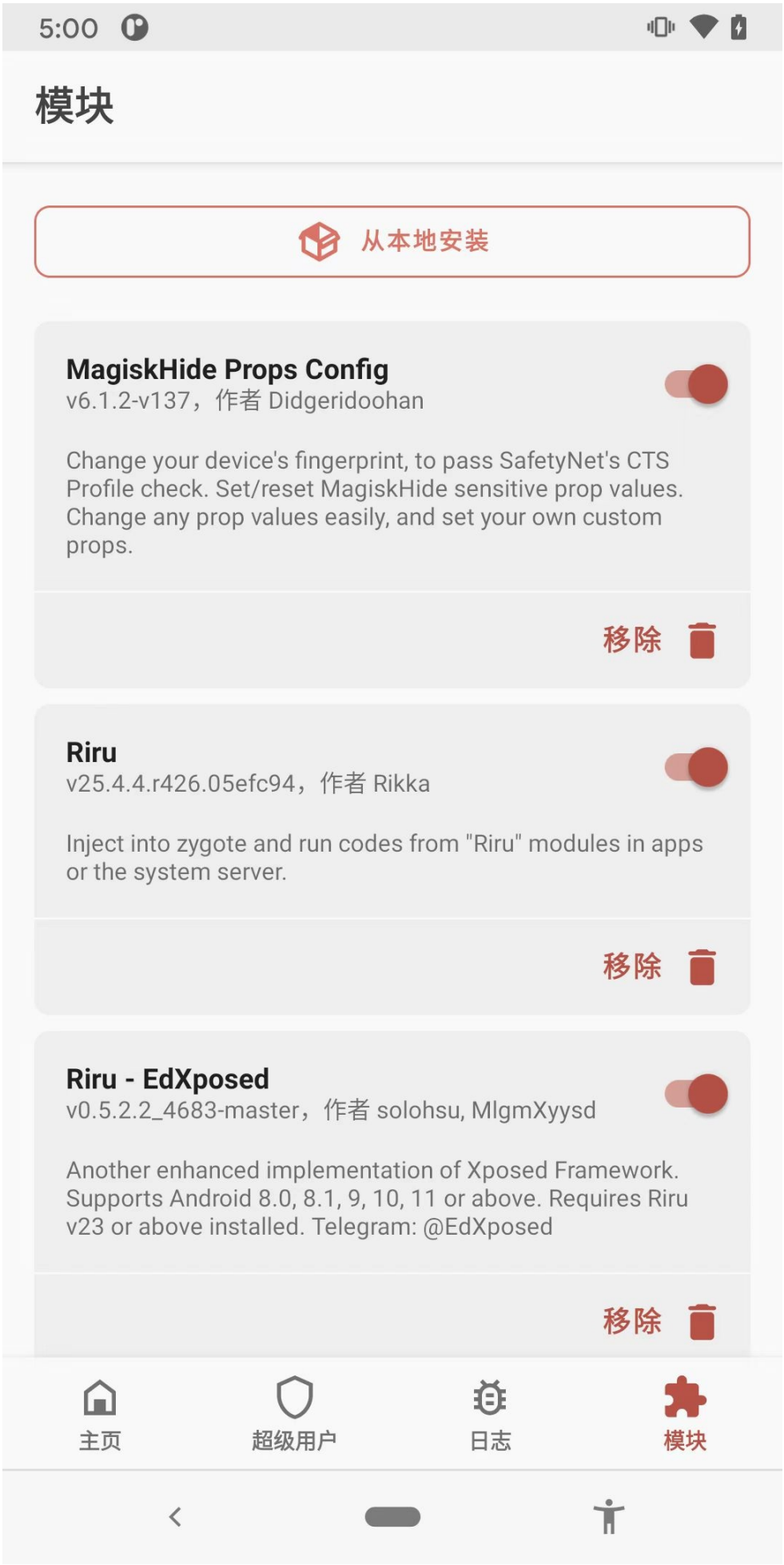

4:02
安装
完成!
- Copying zip to temp directory
- Installing riru-v26.1.7.r530.ab3086ec9f-release.zip
- Current boot slot: _a
- Device is system-as-root
****
Riru
by Rikka, yujincheng08
****
****
Powered by Magisk
****
- Installing from Magisk app
- Installing Riru v26.1.7.r530.ab3086ec9f (Riru API 26
- Magisk version: 25.2 (25200)
- Device sdk: 30
- Device platform: arm64
- Verified META-INF/com/google/android/update-binary
- Verified customize.sh
- Verified verify.sh
- Extracting Magisk files
- Verified module.prop
- Verified post-fs-data.sh
- Verified service.sh
- Verified system.prop
- Verified util_functions.sh
- Verified uninstall.sh
- Extracting arm libraries
- Verified lib/armeabi-v7a/libriru.so
- Verified lib/armeabi-v7a/libriruhide.so
- Verified lib/armeabi-v7a/libriruloader.so
- Extracting arm64 libraries
- Verified lib/arm64-v8a/libriru.so
- Verified lib/arm64-v8a/libriruhide.so
- Verified lib/arm64-v8a/libriruloader.so
- Setting permissions
- Extracting rirud
- Verified rirud.apk
- Checking if your ROM has incorrect SELinux rules
- Start checks...
- No problem found
- Removing old files
- Done
重启

- `EdXposed Manager`

```
adb install EdXposedManager-4.6.2-46200-org.meowcat.edxposed.manager-release.apk
```

安装后效果：

- `Magisk`：新增 `Riru` 和 `Riru - EdXposed` 插件
  -


5:00
模块
从本地安装
MagiskHide Props Config
v6.1.2-v137，作者 Didgeridoohan
Change your device's fingerprint, to pass SafetyNet's CTS Profile check. Set/reset MagiskHide sensitive prop values. Change any prop values easily, and set your own custom props.
移除
Riru
v25.4.4.r426.05efc94，作者 Rikka
Inject into zygote and run codes from "Riru" modules in apps or the system server.
移除
Riru - EdXposed
v0.5.2.2_4683-master，作者 solohsu, MlgmXyysd
Another enhanced implementation of Xposed Framework. Supports Android 8.0, 8.1, 9, 10, 11 or above. Requires Riru v23 or above installed. Telegram: @EdXposed
移除
主页
超级用户
日志
模块

- `EdXposed Manager`： 用于管理EdXposed框架
  -


5:00
EdXposed Manager
EDXPOSED 状态
ALPHA
CANARY
EdXposed 框架已激活
我的设备
v93.0
v0.5.2.2_4683-master_4683 (YAHFA)
v4.6.2 (46200)
Android 11 (Red Velvet Cake, API 30)
Google Google Pixel 3
AArch64 Processor rev 13 (aarch64)
arm64-v8a (arm64)
SELinux 状态: Enforcing
引导验证已启用
引导验证（dm-verity）防止设备在修改系统分区时引导

EdXposed常见问题

请先从 Magisk Manager中安装Riru，Installation failed

- 现象： `Android 11` 的 `Google Pixel 3` 中，最新版 `Magisk` 中去安装最新版 `Riru` 插件： `riru-v26.1.7.r530.ab3086ec9f-release.zip` ，结果报错：

  ```
  ! 请先从 Magisk Manager中安装Riru
    Installation failed
  ```

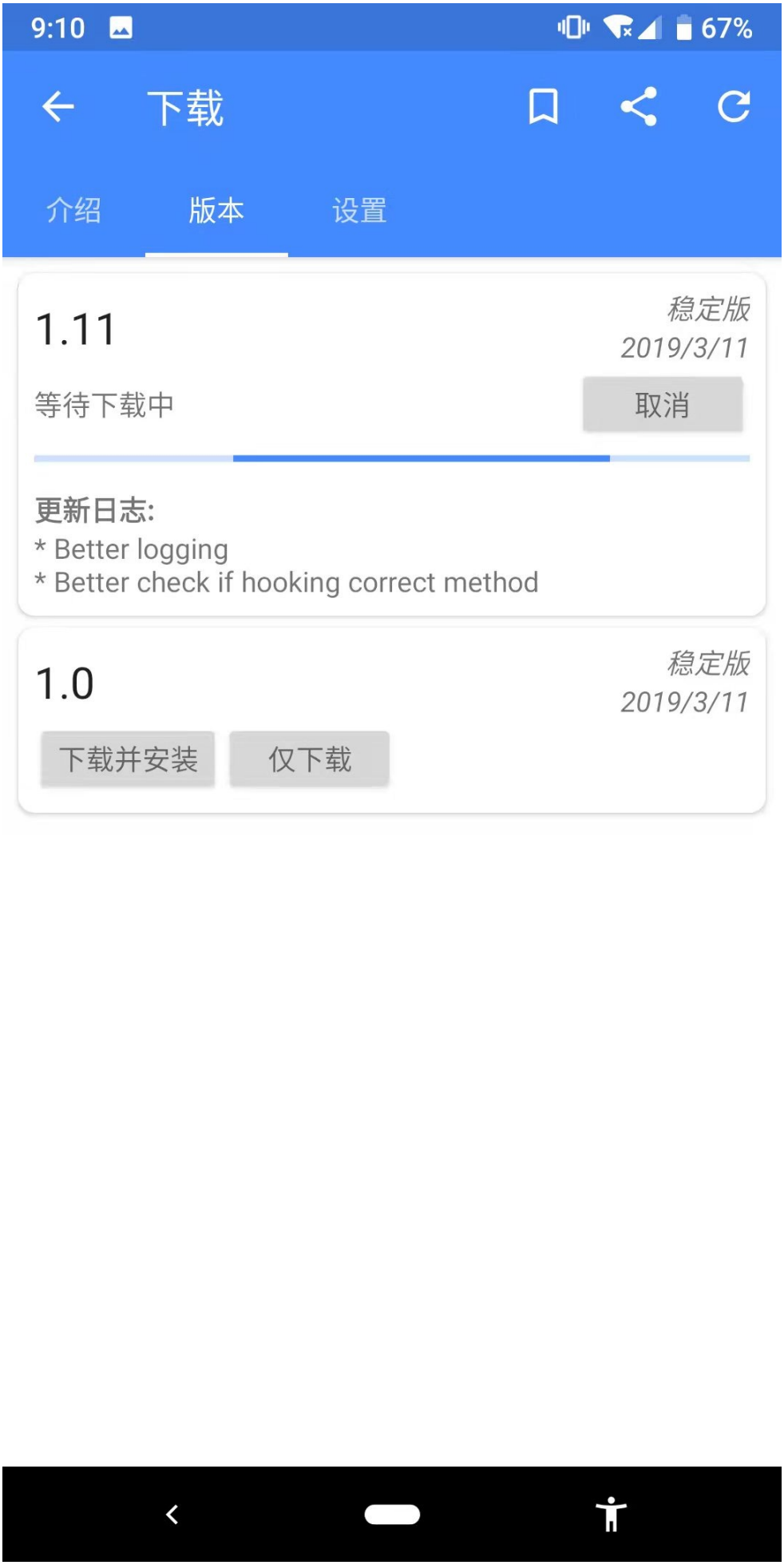
  -


4:06
安装
失败
- Copying zip to temp directory
- Installing EdXposed-v0.5.2.2_4683-master-z-debug.zip
- Current boot slot: _a
- Device is system-as-root
****
Riru - EdXposed
by solohsu, MlgmXyysd
****
****
Powered by Magisk
****
- EdXposed 版本 v0.5.2.2_4683-master
- 已验证 customize.sh
- 已验证 util_functions.sh
- Android SDK 版本: 30
- Magisk 版本: 25200
****
! 未安装
! 请先从 Magisk Manager 中安装 Riru
****
! Installation failed
重启

- 原因
  - 表面原因：已安装的Riru版本，和此处EdXposed不兼容
  - 深层次原因：
    - [BUG] Error detect riru · Issue #873 · ElderDrivers/EdXposed (github.com)
      - The author of Riru is very self-righteous, it discards API compatibility again. For the time being, we don't have the energy to follow an API that is extremely unfriendly to development and compatibility
- 解决办法：换装别的版本的 `Riru v25.4.4`
  - 附录
    - Riru `v25.4.4` 下载地址
      - https://t.me/EdXposed/136
      - Releases · RikkaApps/Riru (github.com)
        - https://github.com/RikkaApps/Riru/releases/download/v25.4.4/riru-v25.4.4-release.zip
        - https://github.com/RikkaApps/Riru/releases/download/v25.4.4/riru-v25.4.4-debug.zip

EdXposed中安装插件卡死显示等待下载中

- 现象：EdXposed中下载插件出现卡死，始终显示：等待下载中，无进度
  -


9:10
67%
下载
介绍
版本
设置
1.11
稳定版
2019/3/11
等待下载中
取消
更新日志:
* Better logging
* Better check if hooking correct method
1.0
稳定版
2019/3/11
下载并安装
仅下载

- 原因：需要翻墙
- 解决办法：去给安卓手机的WiFi加上翻墙代理
  - 此处先拷贝出Mac中的相关代理：

```
export HTTP_PROXY http://127.0.0.1:58591; export HTTPS_PROXY http://127.0.0.1:58591; export ALL_PROXY socks5://127.0.0.1:51837
```

  - 只需要用其中的http的代理即可，端口号是：`58591`
  - 而当前Mac的IP是：`192.168.31.12`

```
ifconfig | grep 192
    inet 192.168.31.12 netmask 0xffffff00 broadcast 192.168.31.255
```

  - 所以去给Pixel3的WiFi的代理配置
    - 参数
      - 代理IP：`192.168.31.12`
      - 代理端口：`58591`
    - 效果
      -


9:21
64%
Xiaomi_R3G_crifan_5G
代理
手动
浏览器会使用 HTTP 代理，但其他应用可能不会使用。
代理服务器主机名
192.168.31.12
代理服务器端口
58591
取消
保存
拼音

- 如此，即可顺利下载插件了
  - 下载完毕后，触发了自动安装
    -


9:24
63%
TrustMeAlready
EdXposed Manager
出于安全考虑，已禁止您的手机安装来自此来源的未知应用。
取消
设置
取消
下一步

安装LSPosed

此处去给 `Android 13` 的 `Google Pixel 5` 安卓手机，去安装 `LSPosed`：

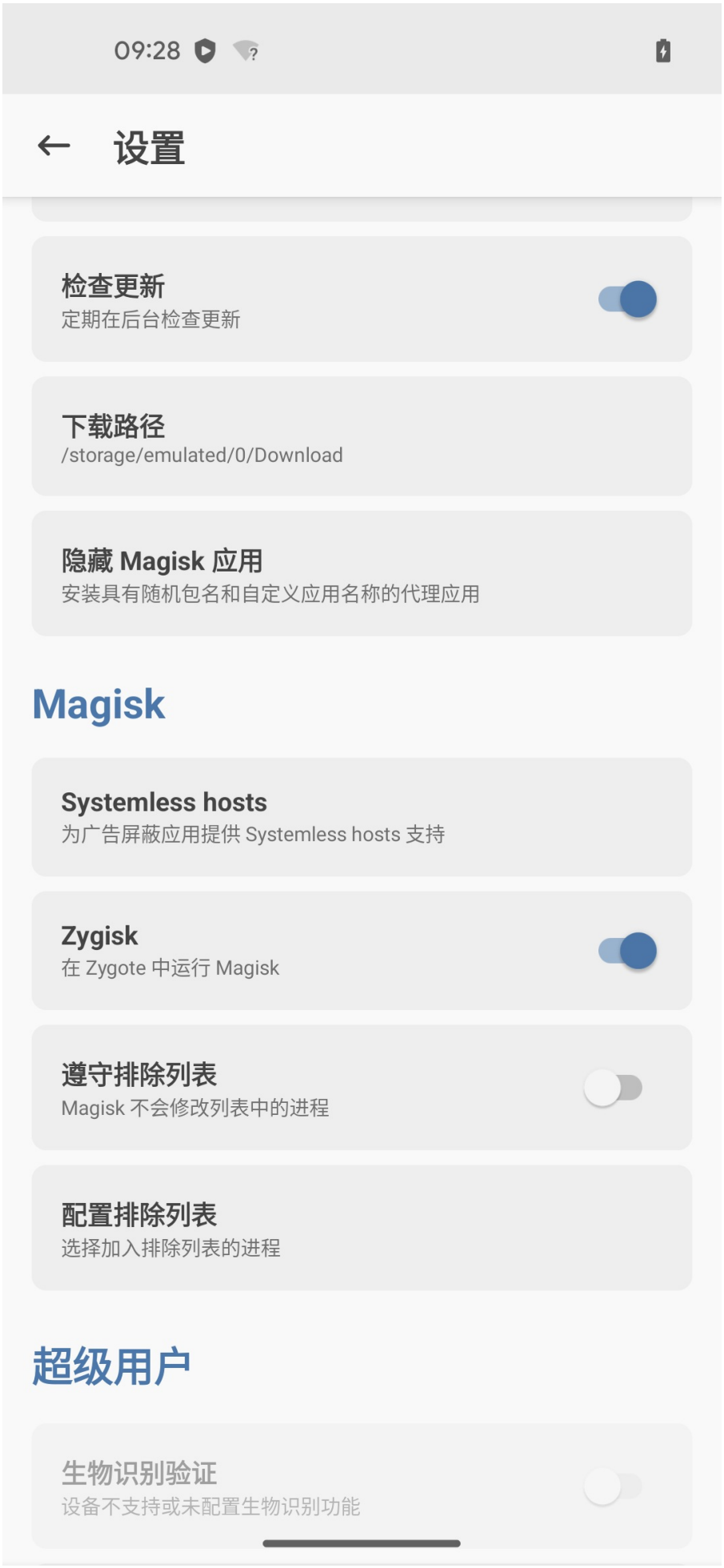
- 安装LSPosed
  - 有2种模式
    - 基于 `Riru` 的
      - 安装方式
        - 很多教程都提到，分2步
          - 先安装 `Riru-Core`
          - 再安装 `Riru-LSPosed`
    - 基于 `Zygisk` 的
      - 安装方式
        - 直接安装：只需一步
          - Magisk中安装插件： `LSPosed-Zygisk`
            - Magisk中勾选：
              - `Zygisk`：在Zygote中运行Magisk

此处详细介绍第二种方式：

Magisk中安装插件：LSPosed-Zygisk

先确保Magisk中：

- 已勾选： `Zygisk`
  -


09:28
设置
检查更新
定期在后台检查更新
下载路径
/storage/emulated/0/Download
隐藏 Magisk 应用
安装具有随机包名和自定义应用名称的代理应用
Magisk
Systemless hosts
为广告屏蔽应用提供 Systemless hosts 支持
Zygisk
在 Zygote 中运行 Magisk
遵守排除列表
Magisk 不会修改列表中的进程
配置排除列表
选择加入排除列表的进程
超级用户
生物识别验证
设备不支持或未配置生物识别功能

然后再去：Magisk中安装 `LSPosed-Zygisk` 插件

概述

- 从LSPosed官网下载到最新版 `LSPosed-v1.8.6-6712-zygisk-release.zip` ，下载到手机中，再去 `Magisk` 中安装该插件，即可

详解

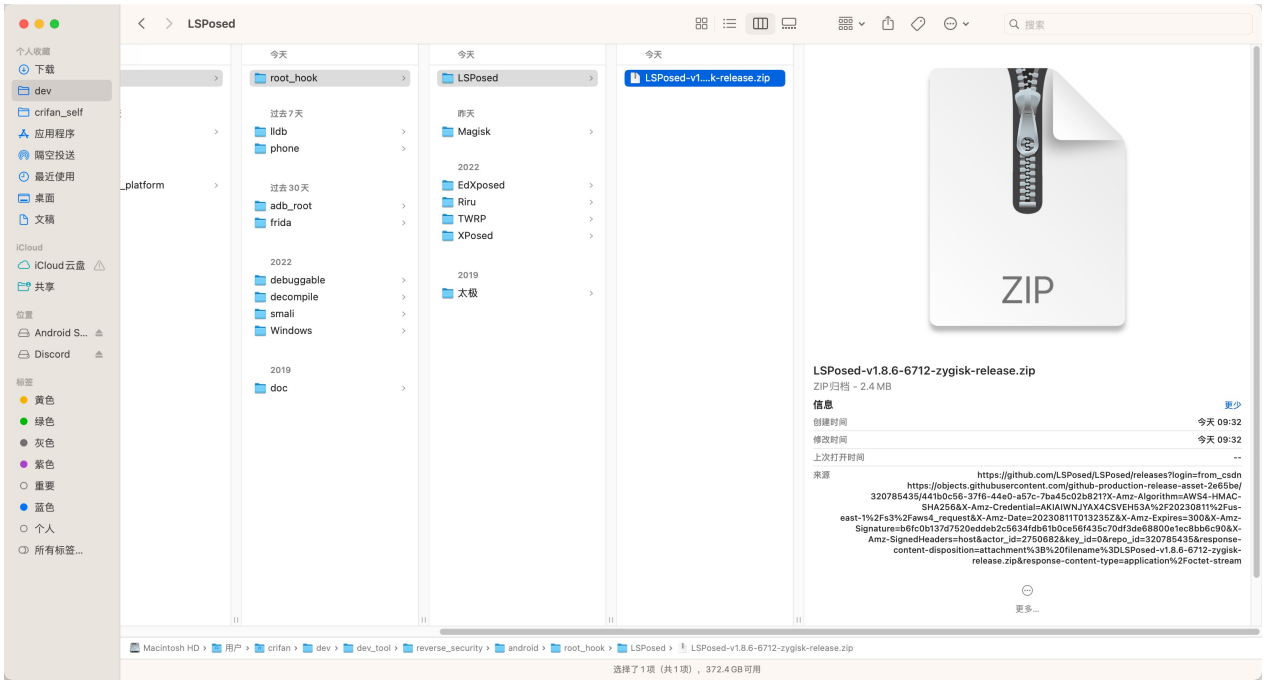
（1）下载LSPosed-Zygisk模块

从LSPosed的官网：Releases · LSPosed/LSPosed (github.com)

找到并下载最新版

https://github.com/LSPosed/LSPosed/releases/download/v1.8.6/LSPosed-v1.8.6-6712-zygisk-release.zip

得到：`LSPosed-v1.8.6-6712-zygisk-release.zip`



（2）用adb下载到安卓手机中

```
adb push LSPosed-v1.8.6-6712-zygisk-release.zip /sdcard/Download/
```

- 输出举例

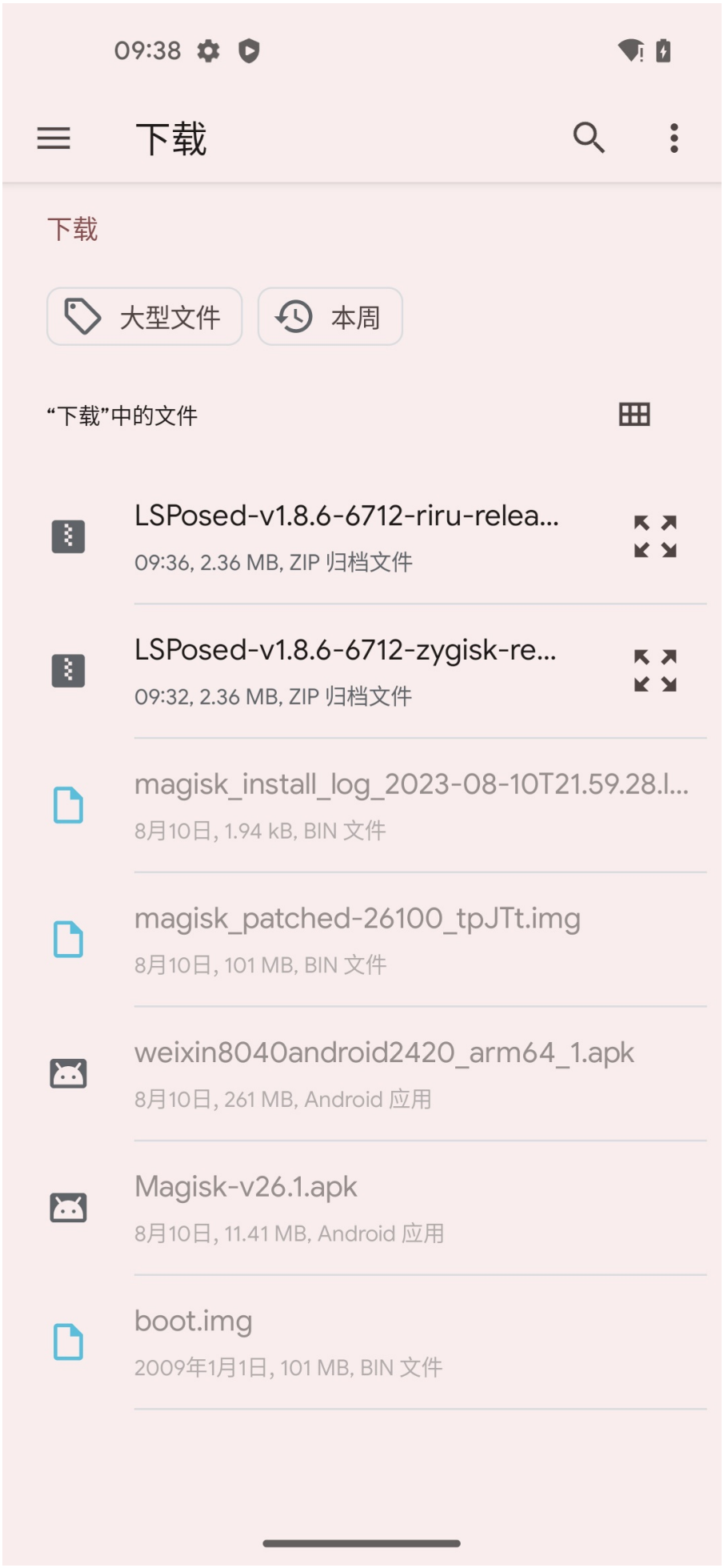
```
➜  LSPosed adb push LSPosed-v1.8.6-6712-zygisk-release.zip /sdcard/Download/
LSPosed-v1.8.6-6712-zygisk-release.zip: 1 file pushed, 0 skipped. 53.5 MB/s (2362560 bytes in 0.042s)
```

（3）Magisk中本地安装LSPosed-Zygisk插件

- `Magisk` -> `模块` -> `从本地安装` ->从下载目录找到并点击 `LSPosed-v1.8.6-6712-zygisk-release.zip` ->安装确认： `确定` -> `重启`
  - `Magisk` -> `模块` -> `从本地安装`
    - ■


09:34
模块
从本地安装
未安装任何模块
主页
超级用户
日志
模块

- 从下载目录找到并点击 `LSPosed-v1.8.6-6712-zygisk-release.zip`
  -


09:38
下载
下载
大型文件
本周
“下载”中的文件
LSPosed-v1.8.6-6712-riru-relea...
09:36, 2.36 MB, ZIP 归档文件
LSPosed-v1.8.6-6712-zygisk-re...
09:32, 2.36 MB, ZIP 归档文件
magisk_install_log_2023-08-10T21.59.28.l...
8月10日, 1.94 kB, BIN 文件
magisk_patched-26100_tpJTt.img
8月10日, 101 MB, BIN 文件
weixin8040android2420_arm64_1.apk
8月10日, 261 MB, Android 应用
Magisk-v26.1.apk
8月10日, 11.41 MB, Android 应用
boot.img
2009年1月1日, 101 MB, BIN 文件

- 安装确认： `确定`
  -

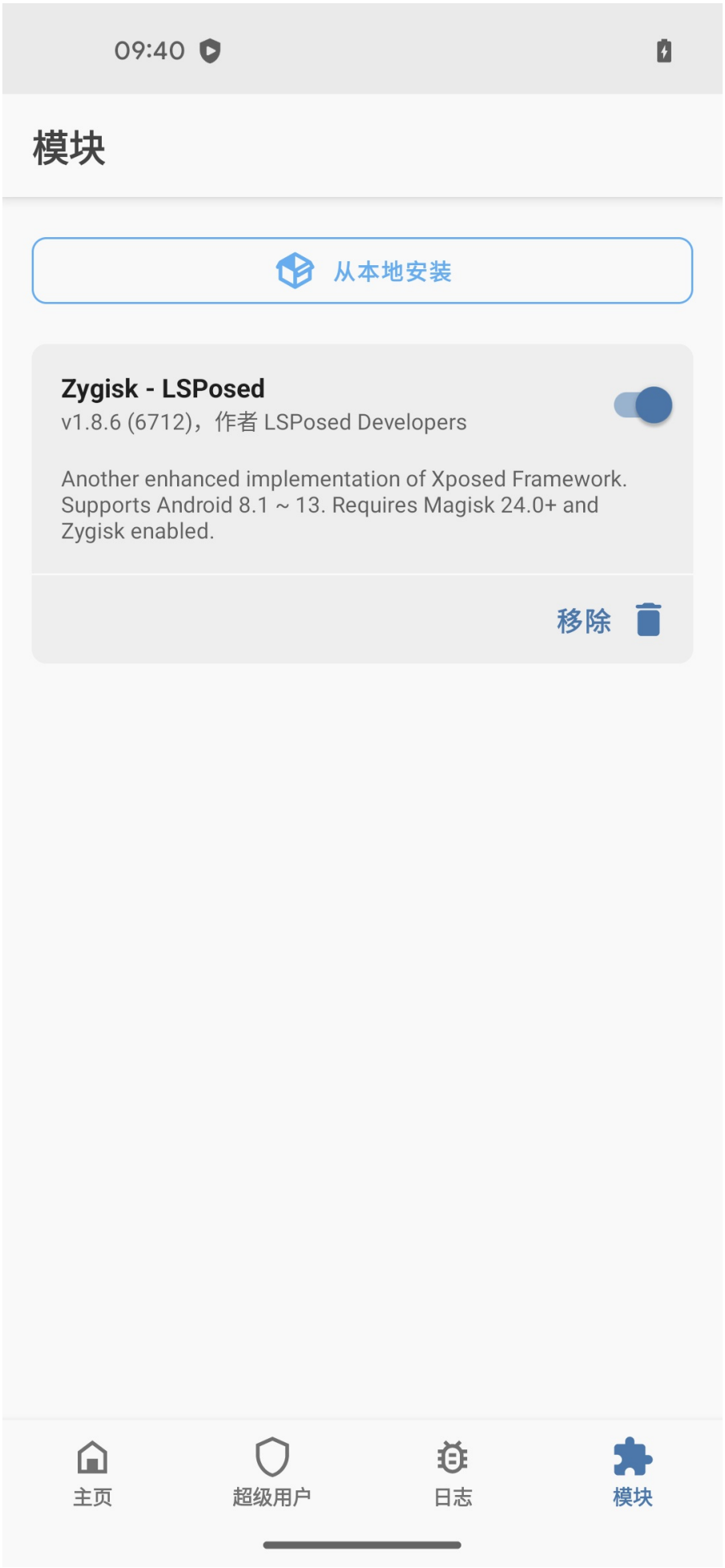

09:38
模块
从本地安装
安装确认
安装模块 LSPosed-v1.8.6-6712-zygisk-release.zip?
取消
确定
主页
超级用户
日志
模块

- 安装完成后，点击 `重启`
  -


09:39
安装
完成!
- Copying zip to temp directory
- Installing LSPosed-v1.8.6-6712-zygisk-release.zip
- Current boot slot: _a
- Device is system-as-root
****
Zygisk - LSPosed
by LSPosed Developers
****
****
Powered by Magisk
****
- LSPosed version v1.8.6 (6712)
- Extracting verify.sh
- Verified META-INF/com/google/android/update-binary
- Verified customize.sh
- Verified verify.sh
- Verified util_functions.sh
- Android SDK version: 33
- Magisk version: 26100
- Installing from Magisk app
- Device platform: arm64
- Extracting module files
- Verified module.prop
- Verified post-fs-data.sh
- Verified service.sh
- Verified uninstall.sh
- Verified framework/lspd.dex
- Verified daemon.apk
- Verified daemon
- Verified manager.apk
- Verified lib/armeabi-v7a/liblspd.so
- Verified lib/arm64-v8a/liblspd.so
- Extracting dex2oat binaries
- Verified bin/armeabi-v7a/dex2oat
- Verified bin/arm64-v8a/dex2oat
- Patching binaries
- Welcome to LSPosed!
- Done
重启

即可在Magisk的模块中看到：

- 已安装： `Zygisk-LSPosed` 模块
  -


09:40
模块
从本地安装
Zygisk - LSPosed
v1.8.6 (6712)，作者 LSPosed Developers
Another enhanced implementation of Xposed Framework. Supports Android 8.1 ~ 13. Requires Magisk 24.0+ and Zygisk enabled.
移除
主页
超级用户
日志
模块

LSPosed相关日志

LSPosed安装过程日志

期间可以点击保存按钮，保存出log日志文件：


09:39
安装
完成!
- Copying zip to temp directory
- Installing LSPosed-v1.8.6-6712-zygisk-release.zip
- Current boot slot: _a
- Device is system-as-root
****
Zygisk - LSPosed
by LSPosed Developers
****
****
Powered by Magisk
****
- LSPosed version v1.8.6 (6712)
- Extracting verify.sh
- Verified META-INF/com/google/android/update-binary
- Verified customize.sh
- Verified verify.sh
- Verified util_functions.sh
- Android SDK version: 33
- Magisk version: 26100
- Installing from Magisk app
- Device platform: arm64
- Extracting module files
- Verified module.prop
- Verified post-fs-data.sh
- Verified service.sh
- Verified uninstall.sh
- Verified framework/lspd.dex
- Verified daemon.apk
- Verified daemon
- Verified manager.apk
- Verified lib/armeabi-v7a/liblspd.so
- Verified lib/arm64-v8a/liblspd.so
- Extracting dex2oat binaries
- Verified bin/armeabi-v7a/dex2oat
- Verified bin/arm64-v8a/dex2oat
- Patching binaries
- Welcome to LSPosed!
- Done
/storage/emulated/0/Download/
magisk_install_log_2023-08-11T09.39.04.log
重启

再去把log日志导出上传到电脑端：

```
➜  LSPosed adb pull /sdcard/Download/magisk_install_log_2023-08-11T09.39.04.log .
/sdcard/Download/magisk_install_log_2023-08-11T09.39.04.log: 1 file pulled, 0 skipped. 0.7 MB/s (4527 bytes in 0.006s)
```

查看到内容是：

- `magisk_install_log_2023-08-11T09.39.04.log`

```
- Copying zip to temp directory
- Installing LSPosed v1.8.6 (6712) zygisk release.zip
- Current boot slot: _a
- Device is system as root
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: module.prop
**********************
 Zygisk - LSPosed
 by LSPosed Developers
**********************
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: customize.sh
*******************
 Powered by Magisk
*******************
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: verify.sh
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
   creating: META-INF/com/google/android/
  inflating: META-INF/com/google/android/update-binary.sha256
  inflating: META-INF/com/google/android/updater-script.sha256
  inflating: META-INF/com/google/android/update-binary
  inflating: META-INF/com/google/android/updater-script
- LSPosed version: v1.8.6 (6712)
- Extracting verify.sh
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: customize.sh
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: customize.sh.sha256
- Verified META-INF/com/google/android/update-binary
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: verify.sh
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: verify.sh.sha256
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: util_functions.sh
- Verified customize.sh
- Verified verify.sh
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: util_functions.sh.sha256
- Verified util_functions.sh
- Android SDK version: 33
- Magisk version: 26100
- Installing from Magisk app
- Device platform: arm64
- Extracting module files
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: module.prop
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: module.prop.sha256
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: post-fs-data.sh
- Verified module.prop
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: post-fs-data.sh.sha256
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: service.sh
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: service.sh.sha256
- Verified post-fs-data.sh
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: uninstall.sh
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
```

```
  inflating: uninstall.sh.sha256
- Verified service.sh
- Verified uninstall.sh
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: framework lspd dex
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: framework lspd dex.sha256
- Verified framework lspd dex
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: daemon.apk
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: daemon.apk.sha256
- Verified daemon.apk
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: daemon
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: daemon.sha256
- Verified daemon
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: manager.apk
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: manager.apk.sha256
- Verified manager.apk
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: liblspd.so
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: liblspd.so.sha256
- Verified lib armeabi v7a liblspd so
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: liblspd.so
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: liblspd.so.sha256
- Verified lib arm64 v8a liblspd so
- Extracting dex2oat binaries
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: dex2oat
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: dex2oat.sha256
- Verified bin armeabi v7a dex2oat
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: dex2oat
Archive:  /data/user/0/com.topjohnwu.magisk/cache/flash/install.zip
  inflating: dex2oat.sha256
- Verified bin arm64 v8a dex2oat
- Patching binaries
- Welcome to LSPosed
- Done
```

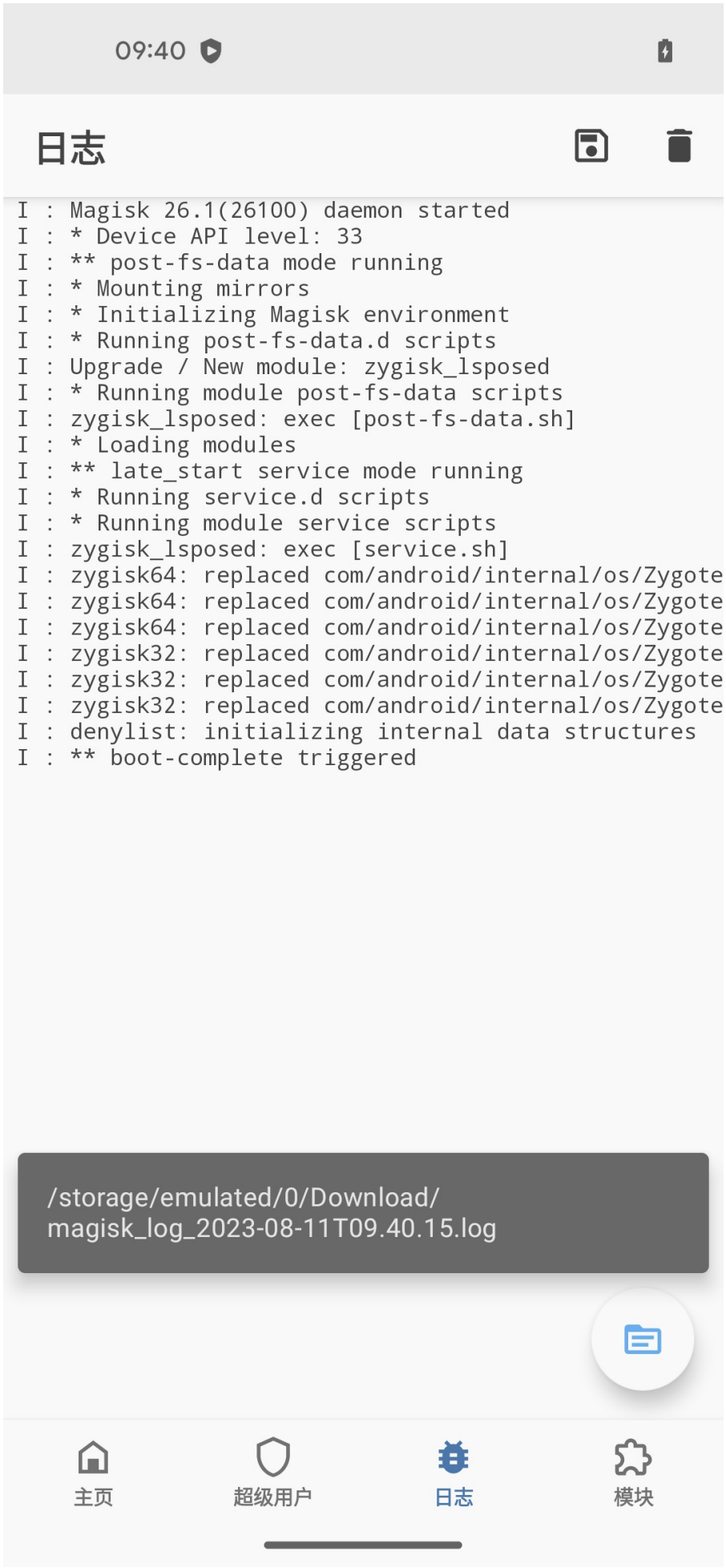
供参考。

Magisk启动日志带LSPosed

Magisk重启后，相关日志中，能看到LSPosed字样：

- `zygisk_lsposed: exec [service.sh]`

也去点击保存按钮，保存出日志文件：


09:40
日志
I : Magisk 26.1(26100) daemon started
I : * Device API level: 33
I : ** post-fs-data mode running
I : * Mounting mirrors
I : * Initializing Magisk environment
I : * Running post-fs-data.d scripts
I : Upgrade / New module: zygisk_lsposed
I : * Running module post-fs-data scripts
I : zygisk_lsposed: exec [post-fs-data.sh]
I : * Loading modules
I : ** late_start service mode running
I : * Running service.d scripts
I : * Running module service scripts
I : zygisk_lsposed: exec [service.sh]
I : zygisk64: replaced com/android/internal/os/Zygote
I : zygisk64: replaced com/android/internal/os/Zygote
I : zygisk64: replaced com/android/internal/os/Zygote
I : zygisk32: replaced com/android/internal/os/Zygote
I : zygisk32: replaced com/android/internal/os/Zygote
I : zygisk32: replaced com/android/internal/os/Zygote
I : denylist: initializing internal data structures
I : ** boot-complete triggered
/storage/emulated/0/Download/
magisk_log_2023-08-11T09.40.15.log
主页
超级用户
日志
模块

导出日志文件：

```
➜  LSPosed adb pull /sdcard/Download/magisk_log_2023-08-11T09.40.15.log .
/sdcard/Download/magisk_log_2023-08-11T09.40.15.log: 1 file pulled, 0 skipped. 11.3 MB/s (96550 bytes in 0.008s)
```

可以看到：log日志内容很多。

此处只摘录相关的核心日志：

```
---Magisk Logs---
26.1 (26100)

03-18 05:03:44.576   964   964 I : Magisk 26.1(26100) daemon started
03-18 05:03:44.578   964   964 I : * Device API level: 33
03-18 05:03:44.587   964   966 I : ** post-fs-data mode running
03-18 05:03:44.587   964   966 I : * Mounting mirrors
03-18 05:03:44.625   964   966 I : * Initializing Magisk environment
03-18 05:03:44.638   964   966 I : * Running post-fs-data.d scripts
03-18 05:03:44.641   964   966 I : Upgrade / New module: zygisk_lsposed
03-18 05:03:44.641   964   966 I : * Running module post-fs-data scripts
03-18 05:03:44.642   974   974 I : zygisk_lsposed: exec [post-fs-data.sh]
03-18 05:03:44.658   964   966 I : * Loading modules
08-11 09:39:34.139   964   966 I : ** late_start service mode running
08-11 09:39:34.139   964   966 I : * Running service.d scripts
08-11 09:39:34.141   964   966 I : * Running module service scripts
08-11 09:39:34.143   964   966 I : zygisk_lsposed: exec [service.sh]
08-11 09:39:34.255  1015  1015 I : zygisk64: replaced com.android.internal.os.Zygote.nativeForkAndSpecialize
08-11 09:39:34.255  1015  1015 I : zygisk64: replaced com.android.internal.os.Zygote.nativeForkSystemServer
08-11 09:39:34.255  1015  1015 I : zygisk64: replaced com.android.internal.os.Zygote.nativeSpecializeAppProcess
08-11 09:39:34.554  1016  1016 I : zygisk32: replaced com.android.internal.os.Zygote.nativeForkAndSpecialize
08-11 09:39:34.554  1016  1016 I : zygisk32: replaced com.android.internal.os.Zygote.nativeForkSystemServer
08-11 09:39:34.554  1016  1016 I : zygisk32: replaced com.android.internal.os.Zygote.nativeSpecializeAppProcess
08-11 09:39:36.110   964   966 I : denylist: initializing internal data structures
08-11 09:39:47.041   964   966 I : ** boot-complete triggered
```

供参考。

打开LSPosed的app

- LSPosed的app=LSPosed Manager=LSPosed管理器
  - 之前是有独立的app的：`LSPosed Manager`
  - 现在：好像无需安装独立的LSPosed Manager的app，而是：
    - 从**系统通知**中打开`LSPosed的app`

从系统通知中打开LSPosed的app

概述

- 安装好LSPosed后，从系统通知中，点击打开LSPosed的app，后续根据提示，点击创建快捷方式（微件），放到系统桌面，作为普通安卓app使用：点击即可打开LSPosed的app

详细步骤

- Android系统通知 -> LSPosed已加载
  -


09:59 8月11日周五
100%
互联网
蓝牙
手电筒
勿扰
Pixel 设置向导
完成 Pixel 设置
只需再执行几个步骤
Android 系统
2
已连接到 USB 调试 点按即可关闭 USB 调试
正在通过 USB 为此设备充电 点按即可查看更...
静音
已开启 Google Play 保护机制 • 19分钟
此功能可保护您的设备免受有害应用的...
百度一下 • 12分钟
页面已可供查看
LSPosed 已加载

- ->点按通知以打开管理器
  -


09:59 8月11日周五
100%
Pixel 设置向导
完成 Pixel 设置
只需再执行几个步骤
Android 系统
2
已连接到 USB 调试 点按即可关闭 USB 调试
正在通过 USB 为此设备充电 点按即可查看更...
静音
已开启 Google Play 保护机制 • 19分钟
此功能可保护您的设备免受有害应用的...
百度一下 • 12分钟
页面已可供查看
LSPosed
LSPosed 已加载
点按通知以打开管理器
管理
全部清除

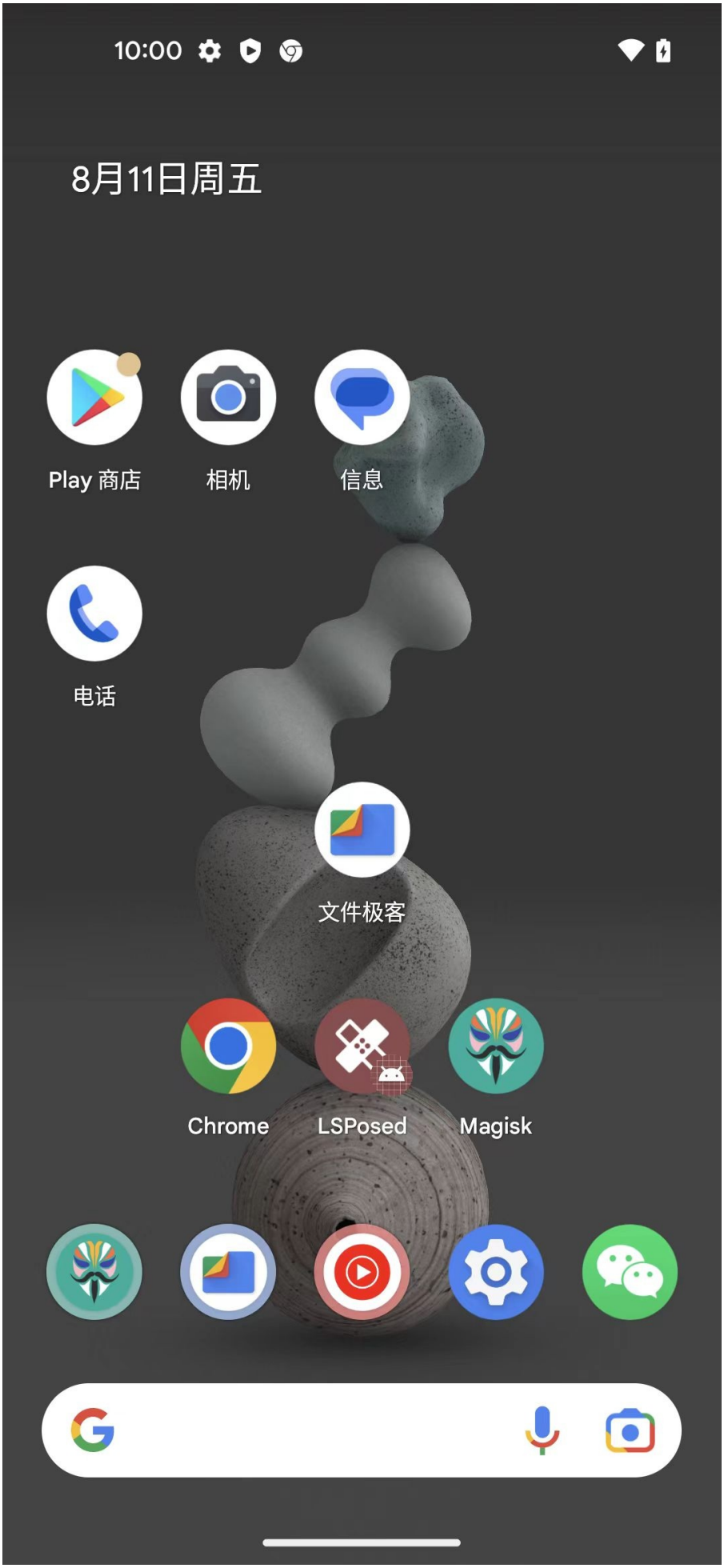
- -> 打开了 LSPosed的app
  - -> 首次打开会出现提示：
    - 欢迎使用LSPosed，你正在使用寄生管理器，可创建快捷方式或继续从通知中打开。 创建快捷方式 不再显示 确定
      -


09:59
LSPosed
已激活
1.8.6 (6712) - Zygisk
欢迎使用 LSPosed
你正在使用寄生管理器，可创建快捷方式或继续从通知中打开。
创建快捷方式
不再显示
确定
框架版本
1.8.6 (6712)
管理器版本
1.8.6 (6712)
系统版本
13 (API 33)
概览

    - -> 点击 创建快捷方式
- -> 出现 Shell弹框，点击：添加到主屏幕
    -


09:59
LSPosed
已激活
1.8.6 (6712) - Zygisk
API 版本
93
Xposed API 调用保护
未启用
Shell
轻触并按住此微件即可在主屏幕上随意移动它
LSPosed
1 × 1
取消
添加到主屏幕

- -> 系统桌面即可出现新图标：LSPosed
  -

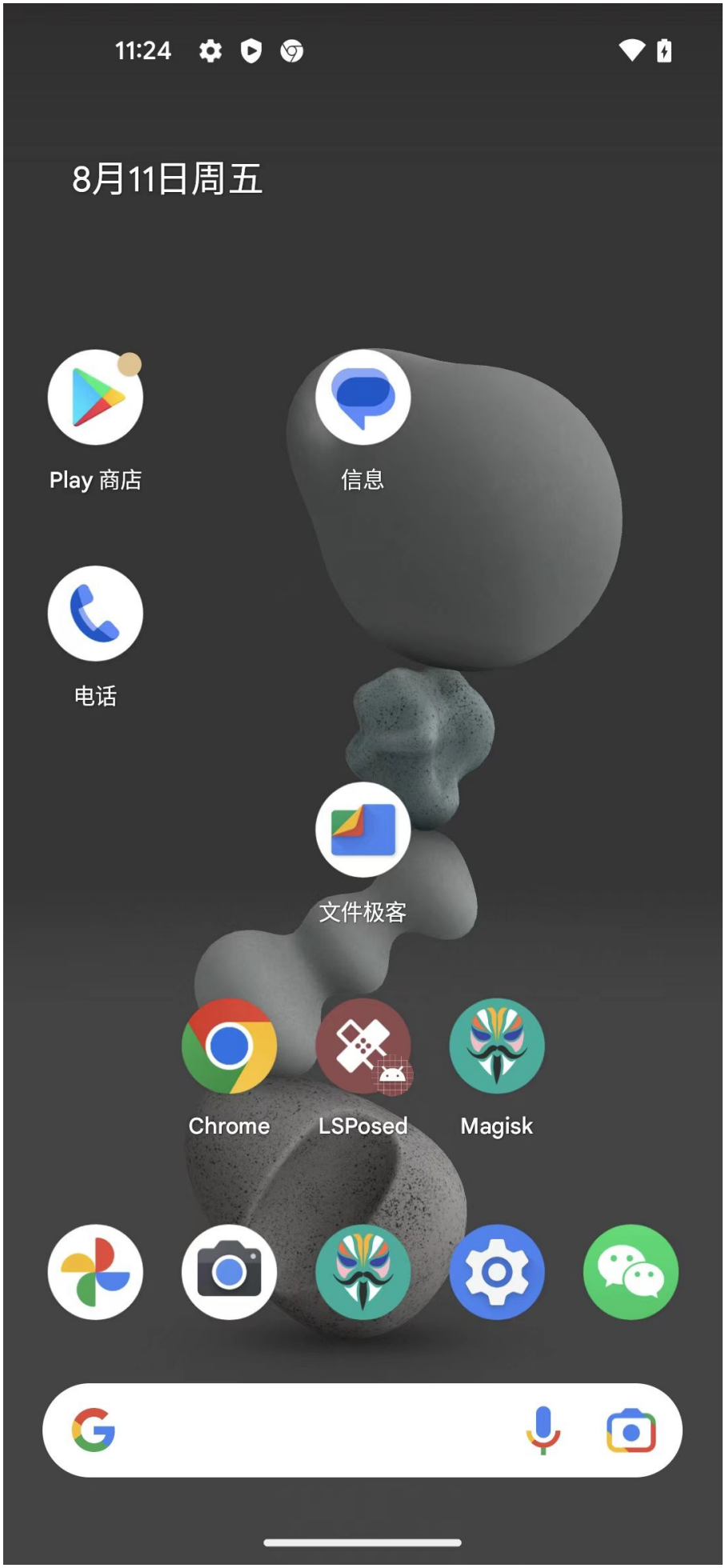

10:00
8月11日周五
Play 商店
相机
信息
电话
文件极客
Chrome
LSPosed
Magisk

    - 注：app的logo图标右下角带个安卓小logo，估计就表示是 微件的意思
- -> 以后即可点击系统桌面图标LSPosed，即可打开启动LSPosed的app
    -


10:00
LSPosed
已激活
1.8.6 (6712) - Zygisk
API 版本
93
Xposed API 调用保护
未启用
Dex 优化器包装
支持
框架版本
1.8.6 (6712)
管理器版本
1.8.6 (6712)
系统版本
13 (API 33)
概览

LSPosed界面和功能

LSPosed桌面logo


11:24
8月11日周五
Play 商店
信息
电话
文件极客
Chrome
LSPosed
Magisk

LSPosed主界面=主页

点击桌面logo，进入，打开主页=主界面：


11:24
LSPosed
已激活
1.8.6 (6712) - Zygisk
API 版本
93
Xposed API 调用保护
未启用
Dex 优化器包装
支持
框架版本
1.8.6 (6712)
管理器版本
1.8.6 (6712)
系统版本
13 (API 33)
概览

下拉可看到更多内容：


11:24
LSPosed
已激活
1.8.6 (6712) - Zygisk
API 版本
93
Xposed API 调用保护
未启用
Dex 优化器包装
支持
框架版本
1.8.6 (6712)
管理器版本
1.8.6 (6712)
系统版本
13 (API 33)
设备
Google Google Pixel 5
系统架构
arm64-v8a
概览

复制内容

可以点击复制：


11:25
LSPosed
API 版本
93
Xposed API 调用保护
未启用
Dex 优化器包装
支持
框架版本
1.8.6 (6712)
管理器版本
1.8.6 (6712)
系统版本
13 (API 33)
设备
Google Google Pixel 5
系统架构
arm64-v8a
复制
已复制到剪贴板
概览

复制出文本内容：

- API 版本：93
- Xposed API 调用保护：未启用
- Dex 优化器包装：支持
- 框架版本：1.8.6 (6712)
- 管理器版本：1.8.6 (6712)
- 系统版本：13 (API 33)
- 设备：Google Google Pixel 5
- 系统架构：arm64-v8a

设置

主页中，右上角 三个点，点击后可以看到：

- 反馈或建议
- 关于

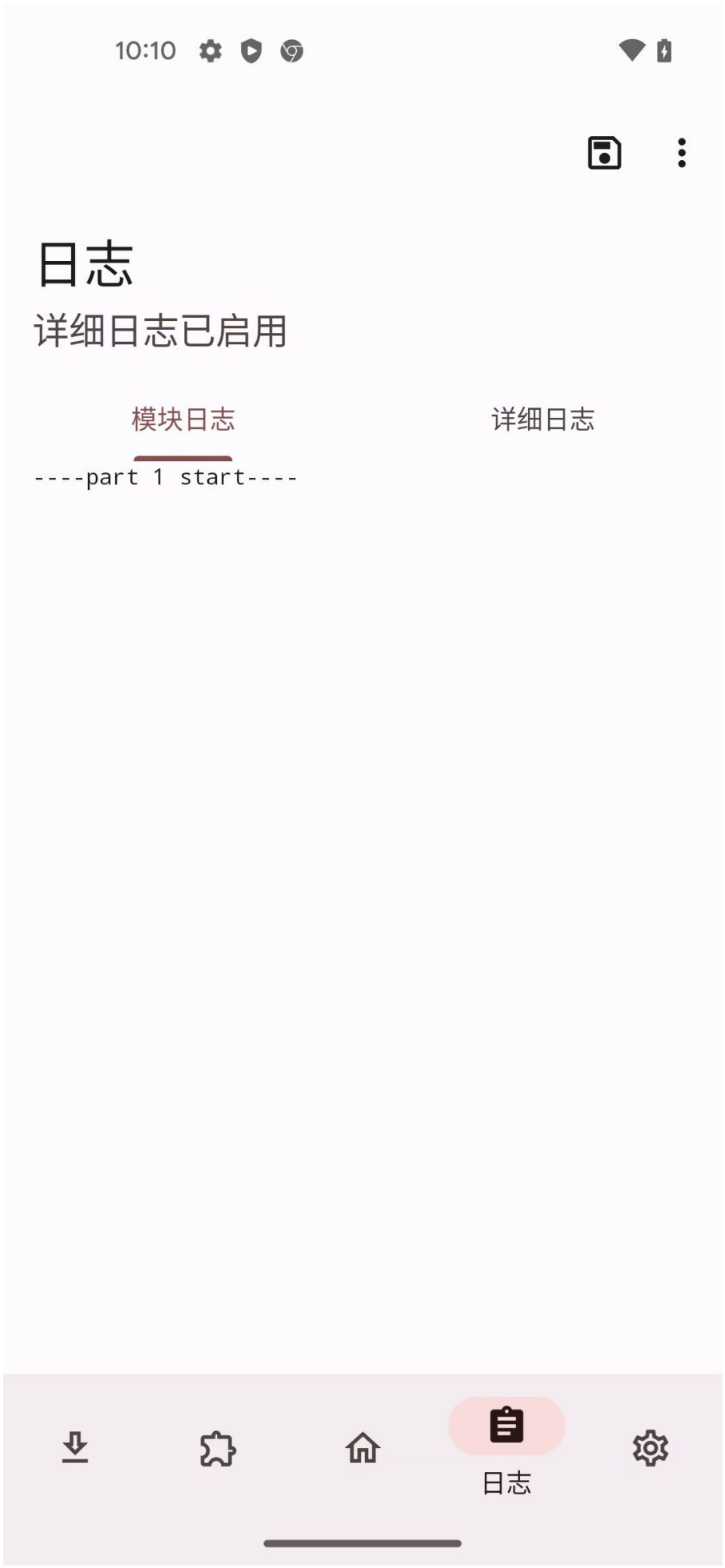

10:09
LSPosed
反馈或建议
关于
API 版本
93
Xposed API 调用保护
未启用
Dex 优化器包装
支持
框架版本
1.8.6 (6712)
管理器版本
1.8.6 (6712)
系统版本
13 (API 33)
设备
Google Google Pixel 5
系统架构
arm64-v8a
复制
概览

关于

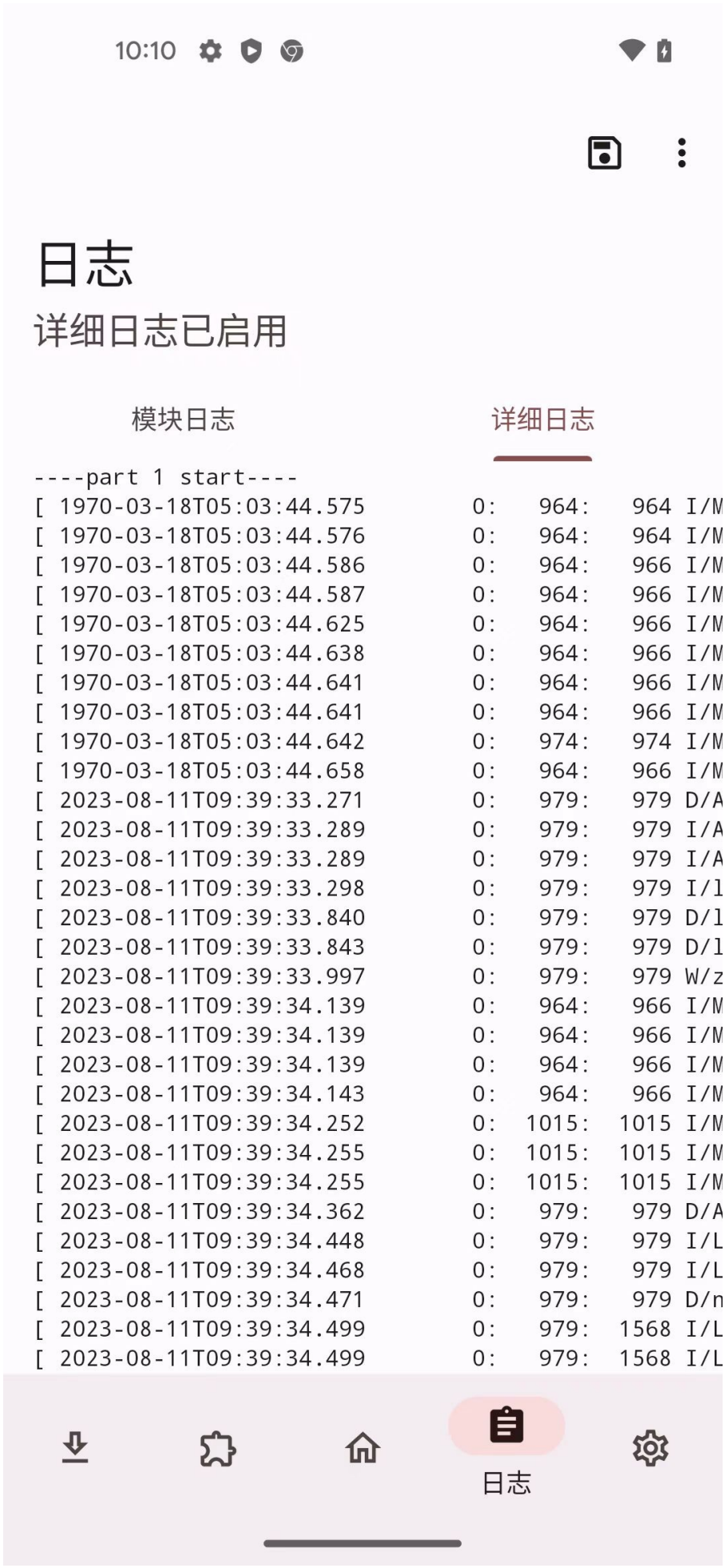

10:09
LSPosed
API 版本
93
Xposed API 调用保护
未启用
Dex 优化器包装
支持
LSPosed
1.8.6 (6712)
在 GitHub 查看源码
加入我们的 Telegram 频道
加入我们的 QQ 频道
设备
Google Google Pixel 5
系统架构
arm64-v8a
复制
概览

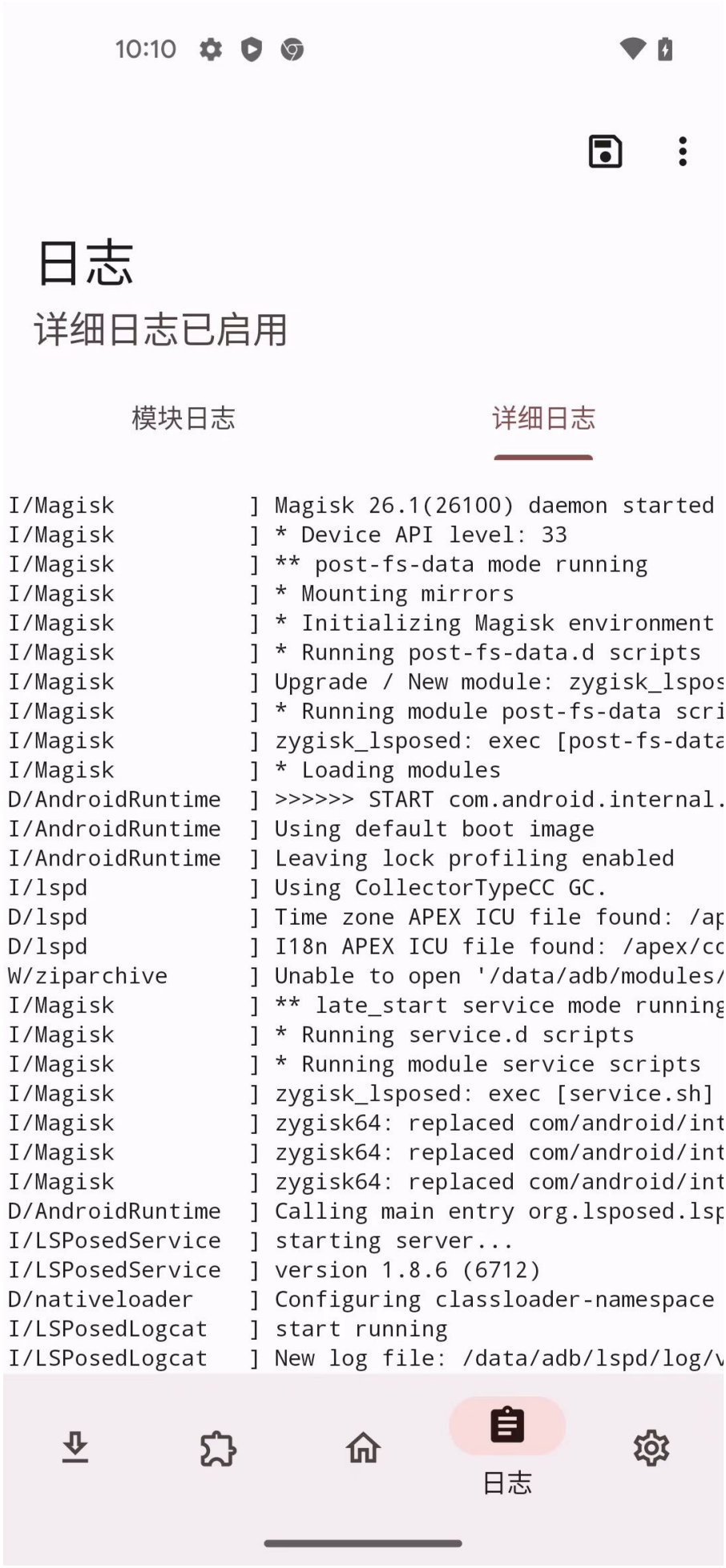
日志

日志 -> 默认显示：模块日志：


10:10
日志
详细日志已启用
模块日志
详细日志
----part 1 start----
日志

点击 切换到 详细日志：


10:10
日志
详细日志已启用
模块日志
详细日志
----part 1 start----
[1970-03-18T05:03:44.575 0: 964: 964 I/M
[1970-03-18T05:03:44.576 0: 964: 964 I/M
[1970-03-18T05:03:44.586 0: 964: 966 I/M
[1970-03-18T05:03:44.587 0: 964: 966 I/M
[1970-03-18T05:03:44.625 0: 964: 966 I/M
[1970-03-18T05:03:44.638 0: 964: 966 I/M
[1970-03-18T05:03:44.641 0: 964: 966 I/M
[1970-03-18T05:03:44.641 0: 964: 966 I/M
[1970-03-18T05:03:44.642 0: 974: 974 I/M
[1970-03-18T05:03:44.658 0: 964: 966 I/M
[2023-08-11T09:39:33.271 0: 979: 979 D/A
[2023-08-11T09:39:33.289 0: 979: 979 I/A
[2023-08-11T09:39:33.289 0: 979: 979 I/A
[2023-08-11T09:39:33.298 0: 979: 979 I/l
[2023-08-11T09:39:33.840 0: 979: 979 D/l
[2023-08-11T09:39:33.843 0: 979: 979 D/l
[2023-08-11T09:39:33.997 0: 979: 979 W/z
[2023-08-11T09:39:34.139 0: 964: 966 I/M
[2023-08-11T09:39:34.139 0: 964: 966 I/M
[2023-08-11T09:39:34.139 0: 964: 966 I/M
[2023-08-11T09:39:34.143 0: 964: 966 I/M
[2023-08-11T09:39:34.252 0: 1015: 1015 I/M
[2023-08-11T09:39:34.255 0: 1015: 1015 I/M
[2023-08-11T09:39:34.255 0: 1015: 1015 I/M
[2023-08-11T09:39:34.362 0: 979: 979 D/A
[2023-08-11T09:39:34.448 0: 979: 979 I/L
[2023-08-11T09:39:34.468 0: 979: 979 I/L
[2023-08-11T09:39:34.471 0: 979: 979 D/n
[2023-08-11T09:39:34.499 0: 979: 1568 I/L
[2023-08-11T09:39:34.499 0: 979: 1568 I/L
日志


10:10
日志
详细日志已启用
模块日志
详细日志
I/Magisk] Magisk 26.1(26100) daemon started
I/Magisk] * Device API level: 33
I/Magisk] ** post-fs-data mode running
I/Magisk] * Mounting mirrors
I/Magisk] * Initializing Magisk environment
I/Magisk] * Running post-fs-data.d scripts
I/Magisk] Upgrade / New module: zygisk_lspos
I/Magisk] * Running module post-fs-data scri
I/Magisk] zygisk_lsposed: exec [post-fs-data
I/Magisk] * Loading modules
D/AndroidRuntime] >>>>>> START com.android.internal.
I/AndroidRuntime] Using default boot image
I/AndroidRuntime] Leaving lock profiling enabled
I/lspd] Using CollectorTypeCC GC.
D/lspd] Time zone APEX ICU file found: /ap
D/lspd] I18n APEX ICU file found: /apex/co
W/ziparchive] Unable to open '/data/adb/modules/
I/Magisk] ** late_start service mode running
I/Magisk] * Running service.d scripts
I/Magisk] * Running module service scripts
I/Magisk] zygisk_lsposed: exec [service.sh]
I/Magisk] zygisk64: replaced com/android/int
I/Magisk] zygisk64: replaced com/android/int
I/Magisk] zygisk64: replaced com/android/int
D/AndroidRuntime] Calling main entry org.lsposed.lsp
I/LSPosedService] starting server...
I/LSPosedService] version 1.8.6 (6712)
D/nativeloader] Configuring classloader-namespace
I/LSPosedLogcat] start running
I/LSPosedLogcat] New log file: /data/adb/lspd/log/v
日志

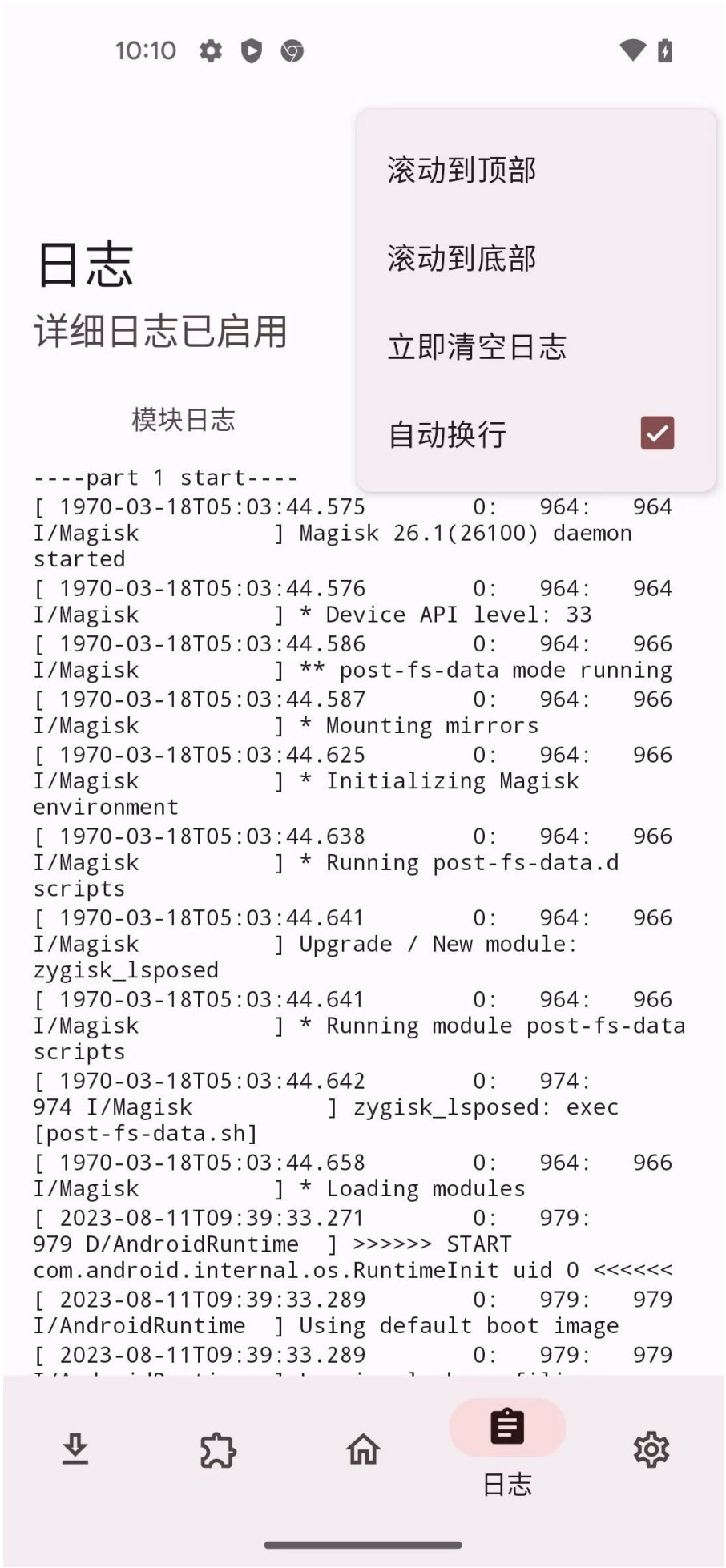
日志选项

继续日志页面，右上角三个点，点击后可以看到各种设置：

- 滚动到顶部
- 滚动到底部
- 立即清空日志
- 自动换行


10:10
滚动到顶部
滚动到底部
立即清空日志
自动换行
日志
详细日志已启用
模块日志
I/Magisk] Magisk 26.1(26100) daemon started
I/Magisk] * Device API level: 33
I/Magisk] ** post-fs-data mode running
I/Magisk] * Mounting mirrors
I/Magisk] * Initializing Magisk environment
I/Magisk] * Running post-fs-data.d scripts
I/Magisk] Upgrade / New module: zygisk_lspos
I/Magisk] * Running module post-fs-data scri
I/Magisk] zygisk_lsposed: exec [post-fs-data
I/Magisk] * Loading modules
D/AndroidRuntime] >>>>>> START com.android.internal.
I/AndroidRuntime] Using default boot image
I/AndroidRuntime] Leaving lock profiling enabled
I/lspd] Using CollectorTypeCC GC.
D/lspd] Time zone APEX ICU file found: /ap
D/lspd] I18n APEX ICU file found: /apex/co
W/ziparchive] Unable to open '/data/adb/modules/
I/Magisk] ** late_start service mode running
I/Magisk] * Running service.d scripts
I/Magisk] * Running module service scripts
I/Magisk] zygisk_lsposed: exec [service.sh]
I/Magisk] zygisk64: replaced com/android/int
I/Magisk] zygisk64: replaced com/android/int
I/Magisk] zygisk64: replaced com/android/int
D/AndroidRuntime] Calling main entry org.lsposed.lsp
I/LSPosedService] starting server...
I/LSPosedService] version 1.8.6 (6712)
D/nativeloader] Configuring classloader-namespace
I/LSPosedLogcat] start running
I/LSPosedLogcat] New log file: /data/adb/lspd/log/v
日志

-》 勾选：自动换行 后的效果：


10:10
滚动到顶部
滚动到底部
立即清空日志
自动换行
日志
详细日志已启用
模块日志
----part 1 start----
[1970-03-18T05:03:44.575 0: 964: 964
I/Magisk] Magisk 26.1(26100) daemon
started
[1970-03-18T05:03:44.576 0: 964: 964
I/Magisk] * Device API level: 33
[1970-03-18T05:03:44.586 0: 964: 966
I/Magisk] ** post-fs-data mode running
[1970-03-18T05:03:44.587 0: 964: 966
I/Magisk] * Mounting mirrors
[1970-03-18T05:03:44.625 0: 964: 966
I/Magisk] * Initializing Magisk
environment
[1970-03-18T05:03:44.638 0: 964: 966
I/Magisk] * Running post-fs-data.d
scripts
[1970-03-18T05:03:44.641 0: 964: 966
I/Magisk] Upgrade / New module:
zygisk_lsposed
[1970-03-18T05:03:44.641 0: 964: 966
I/Magisk] * Running module post-fs-data
scripts
[1970-03-18T05:03:44.642 0: 974:
974 I/Magisk] zygisk_lsposed: exec
[post-fs-data.sh]
[1970-03-18T05:03:44.658 0: 964: 966
I/Magisk] * Loading modules
[2023-08-11T09:39:33.271 0: 979:
979 D/AndroidRuntime] >>>>>> START
com.android.internal.os.RuntimeInit uid 0 <<<<<<
[2023-08-11T09:39:33.289 0: 979: 979
I/AndroidRuntime] Using default boot image
[2023-08-11T09:39:33.289 0: 979: 979
日志

仓库

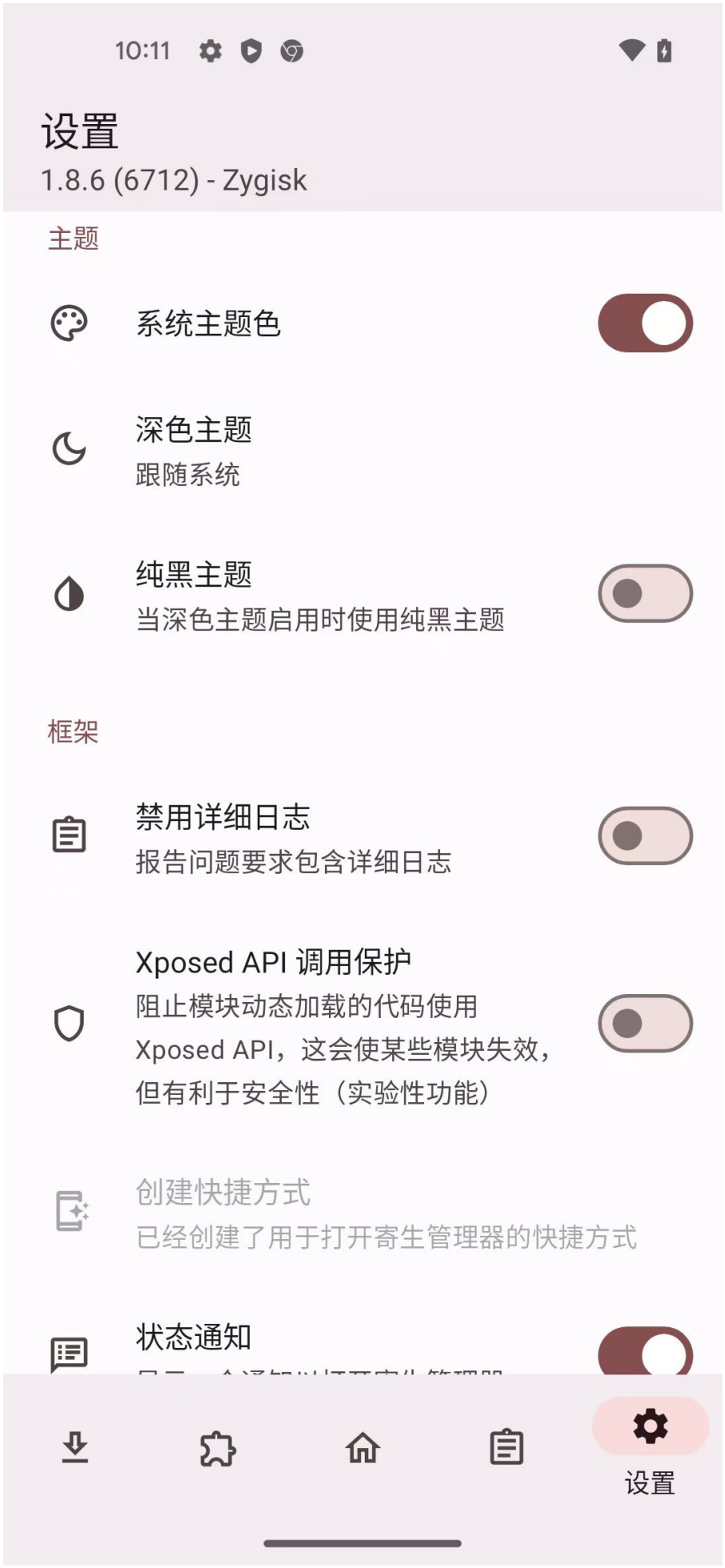

10:09
仓库
所有模块均已最新
555电影去广告
com.kgg.hhvideo
555电影去广告插件
更新于 2023/4/29
爱美剧模块
com.tcgg.imjhook
爱美剧永久会员
更新于 2023/7/2
哔哩发评反诈
icu.freedomintrovert.bilisendcommantifraud
哔哩哔哩发送评论自动检查，恢复在阿瓦隆系统下你对评论状态的知情权!
更新于 2023/6/27
哔哩漫游/BiliRoaming
me.iacn.biliroaming
解除B站客户端番剧区域限制，并且提供其他小功能
Unblock bangumi area limit of BILIBILI, and miscellaneous features
仓库

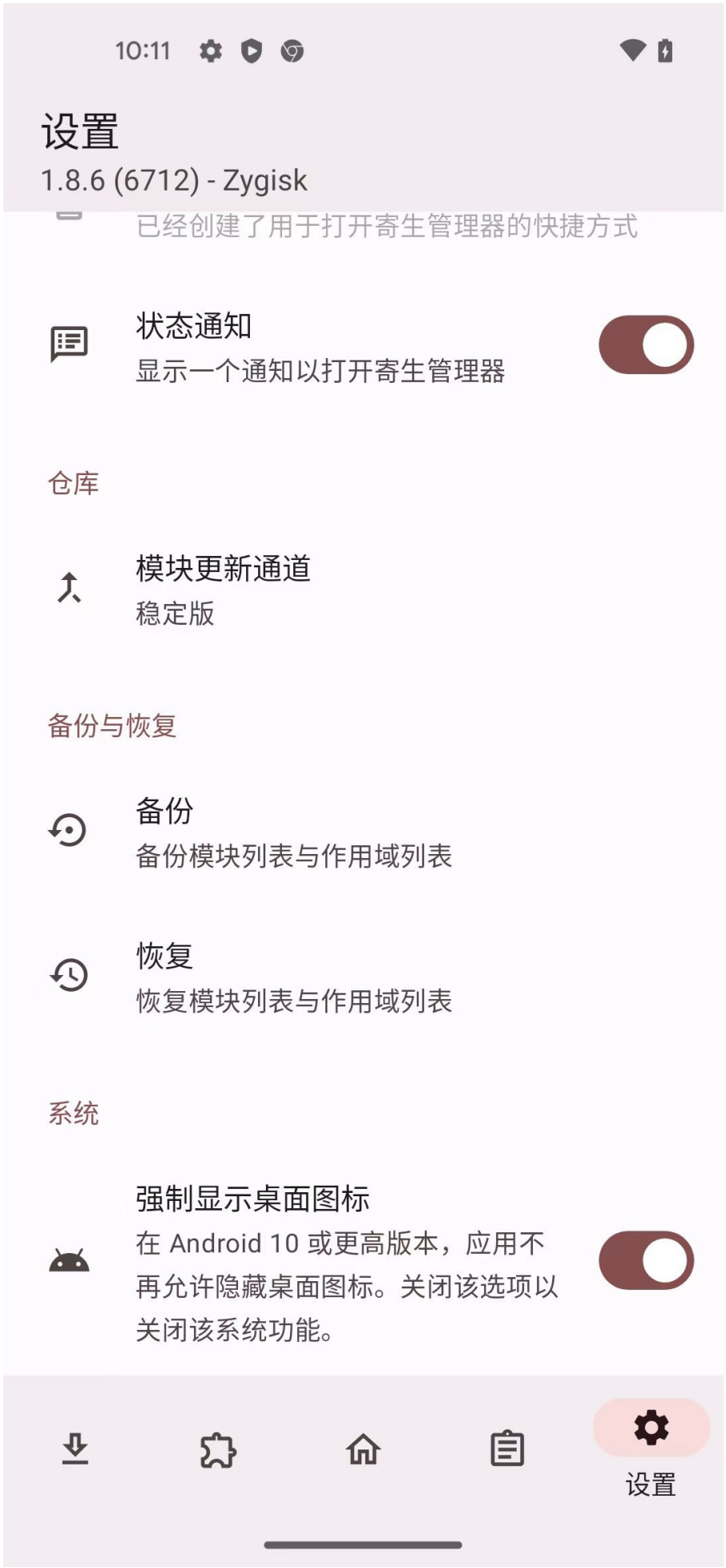
模块


10:09
模块
已启用 0 个模块
¯_(ツ)_/¯
空空如也
模块

设置


10:11
设置
1.8.6 (6712) - Zygisk
网络
安全 DNS（DoH）
解决某些地区的 DNS 污染问题
语言
语言
跟随系统
译者
LSPosed
参与翻译
帮助我们把 LSPosed 翻译到你的语言
主题
系统主题色
设置

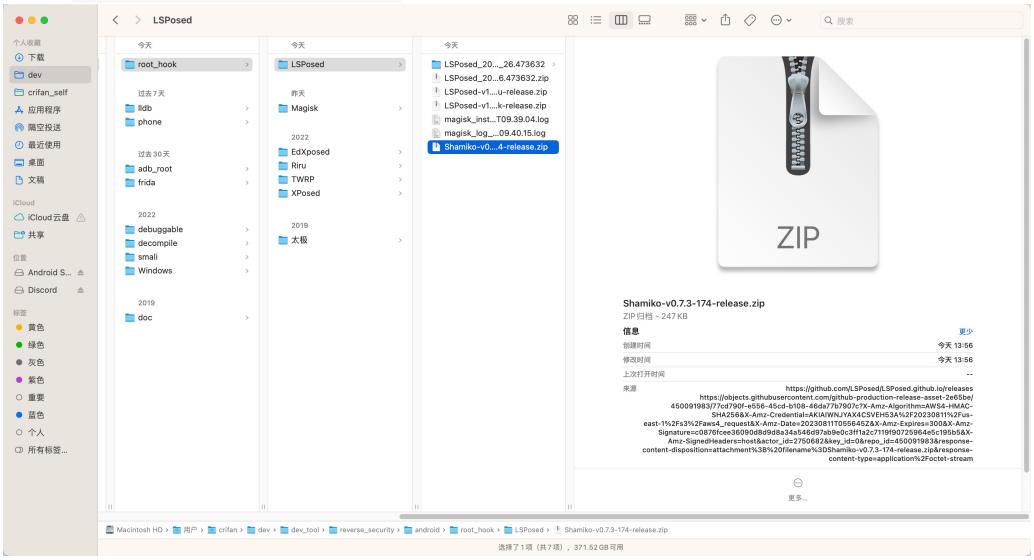

10:11
设置
1.8.6 (6712) - Zygisk
主题
系统主题色
深色主题
跟随系统
纯黑主题
当深色主题启用时使用纯黑主题
框架
禁用详细日志
报告问题要求包含详细日志
Xposed API 调用保护
阻止模块动态加载的代码使用
Xposed API，这会使某些模块失效，
但有利于安全性（实验性功能）
创建快捷方式
已经创建了用于打开寄生管理器的快捷方式
状态通知
设置


10:11
设置
1.8.6 (6712) - Zygisk
已经创建了用于打开寄生管理器的快捷方式
状态通知
显示一个通知以打开寄生管理器
仓库
模块更新通道
稳定版
备份与恢复
备份
备份模块列表与作用域列表
恢复
恢复模块列表与作用域列表
系统
强制显示桌面图标
在 Android 10 或更高版本，应用不再允许隐藏桌面图标。关闭该选项以关闭该系统功能。
设置

MagiskHide替代方案：Shamiko模块

- 背景：
  - 之前Magisk有个 `MagiskHide` 的模块
    - 功能：实现隐藏root -> 避免别的app检测到安卓已root
      - 功能很好用
  - 但是由于Magisk `24.0+` 移除了 `MagiskHide`，因此提供一个替代的隐藏root的方案：`Shamiko模块`

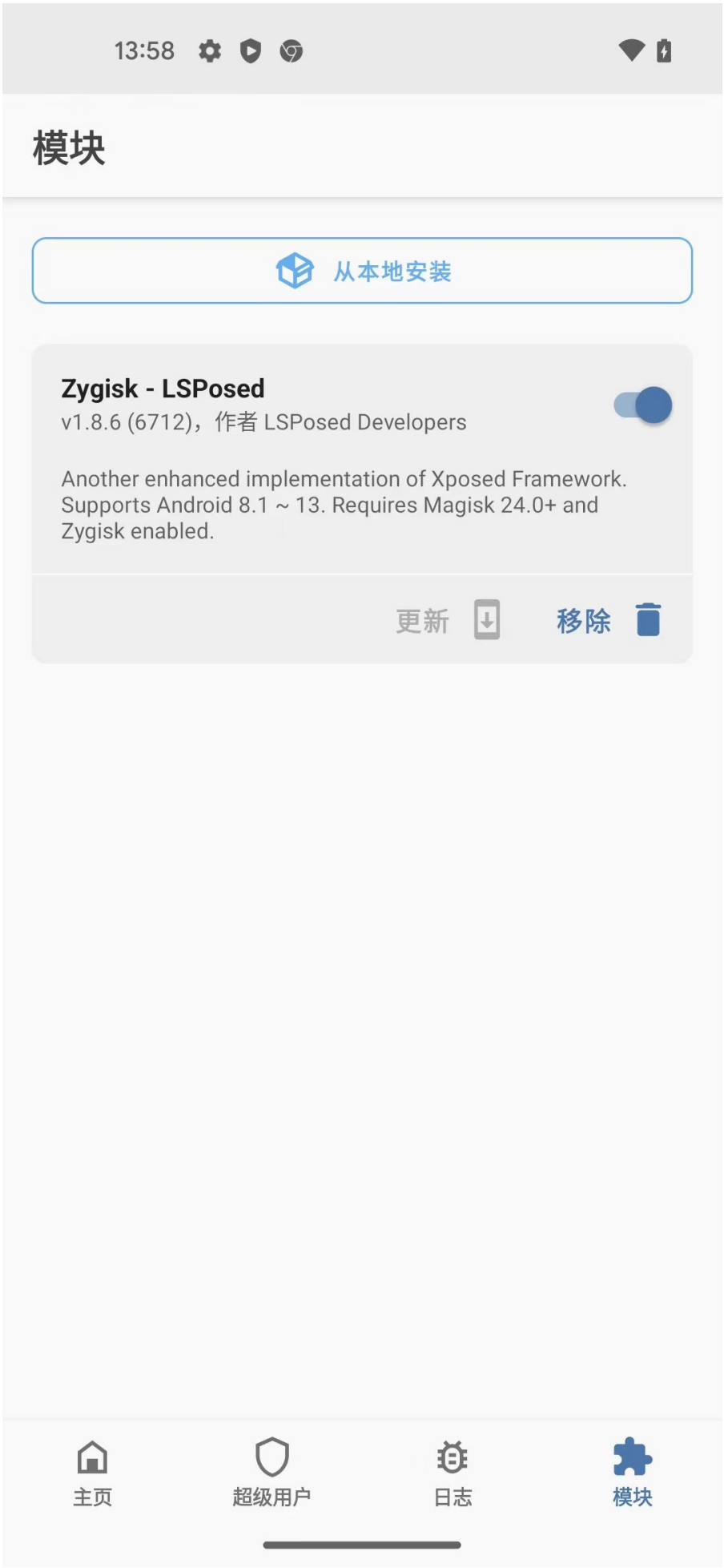
下载和安装 `Shamiko` 模块

- 下载到电脑
  - LSPosed Official Website | LSPosed.github.io
    - Releases · LSPosed/LSPosed.github.io
      - 找到最新版
        - https://github.com/LSPosed/LSPosed.github.io/releases/download/shamiko-174/Shamiko-v0.7.3-174-release.zip
      - 下载得到：`Shamiko-v0.7.3-174-release.zip`
      - 

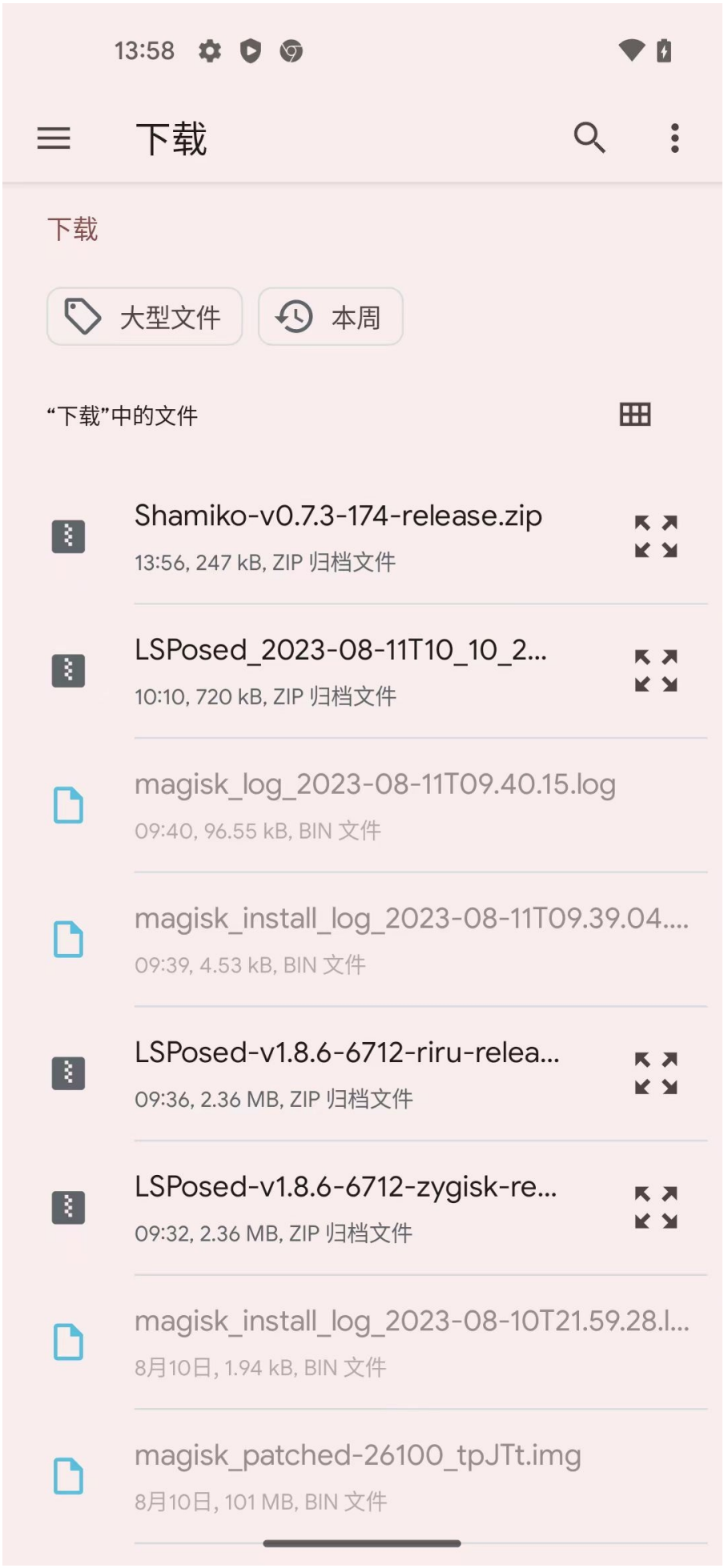
- 传输下载到安卓手机

```
➜  LSPosed adb push Shamiko-v0.7.3-174-release.zip /sdcard/Download/
Shamiko-v0.7.3-174-release.zip: 1 file pushed, 0 skipped. 327.6 MB/s (247141 bytes in 0.001s)
```

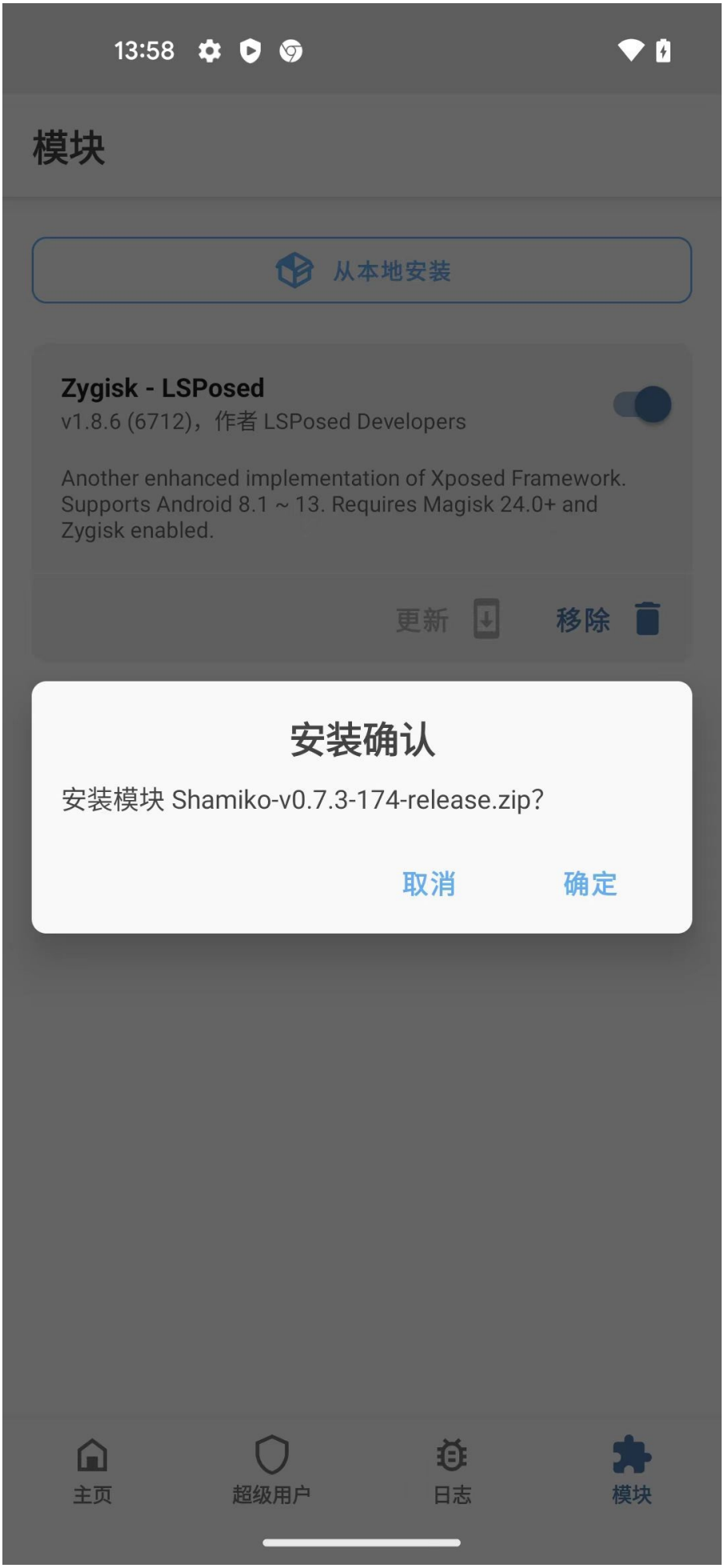
- 去 `Magisk` 中安装Shamiko模块
  - 打开Magisk->模块->从本地安装
    -


13:58
模块
从本地安装
Zygisk - LSPosed
v1.8.6 (6712)，作者 LSPosed Developers
Another enhanced implementation of Xposed Framework. Supports Android 8.1 ~ 13. Requires Magisk 24.0+ and Zygisk enabled.
更新
移除
主页
超级用户
日志
模块

- 点选 `Shamiko-v0.7.3-174-release.zip`
  -


13:58
下载
下载
大型文件
本周
“下载”中的文件
Shamiko-v0.7.3-174-release.zip
13:56, 247 kB, ZIP 归档文件
LSPosed_2023-08-11T10_10_2...
10:10, 720 kB, ZIP 归档文件
magisk_log_2023-08-11T09.40.15.log
09:40, 96.55 kB, BIN 文件
magisk_install_log_2023-08-11T09.39.04....
09:39, 4.53 kB, BIN 文件
LSPosed-v1.8.6-6712-riru-relea...
09:36, 2.36 MB, ZIP 归档文件
LSPosed-v1.8.6-6712-zygisk-re...
09:32, 2.36 MB, ZIP 归档文件
magisk_install_log_2023-08-10T21.59.28.l...
8月10日, 1.94 kB, BIN 文件
magisk_patched-26100_tpJTt.img
8月10日, 101 MB, BIN 文件

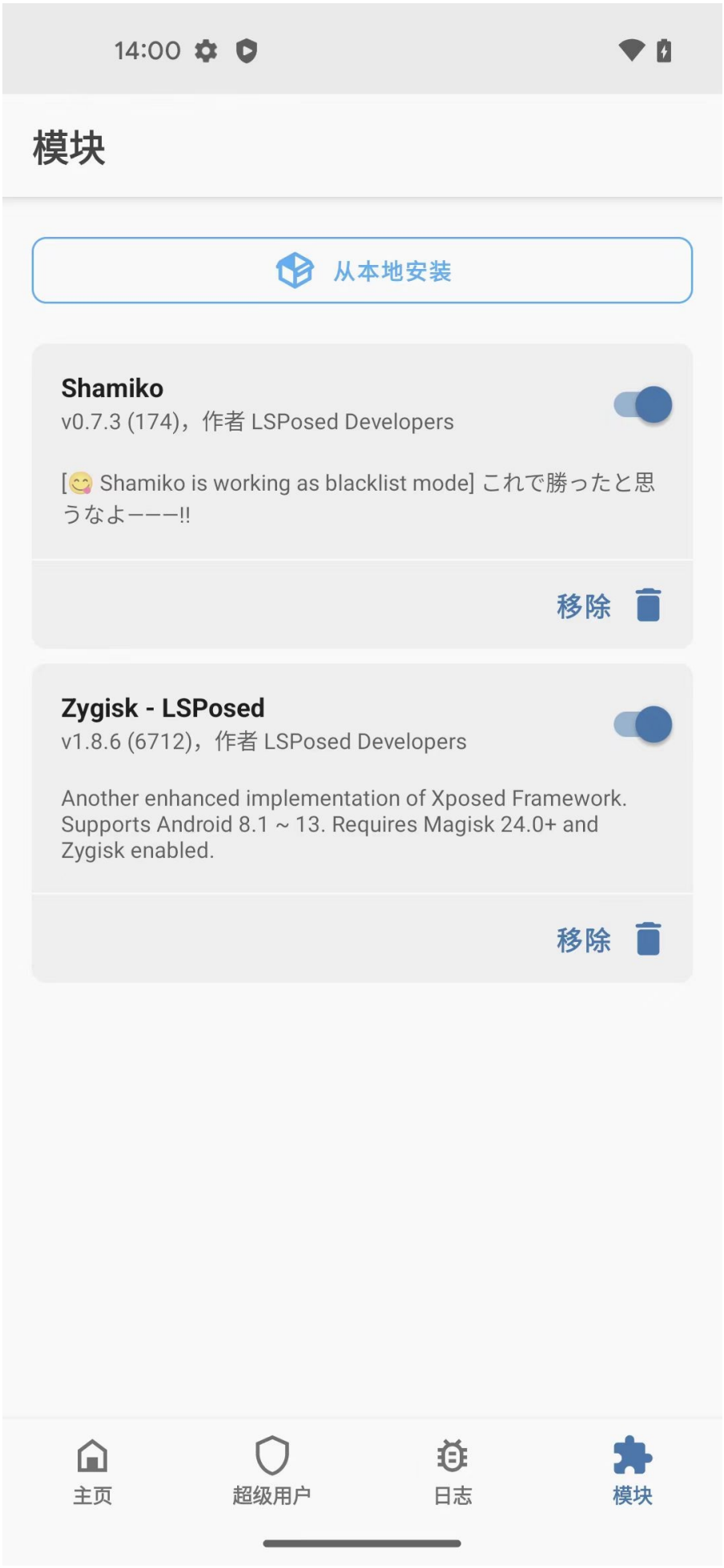
- 安全确认：确定
  -


13:58
模块
从本地安装
Zygisk - LSPosed
v1.8.6 (6712)，作者 LSPosed Developers
Another enhanced implementation of Xposed Framework. Supports Android 8.1 ~ 13. Requires Magisk 24.0+ and Zygisk enabled.
更新
移除
安装确认
安装模块 Shamiko-v0.7.3-174-release.zip?
取消
确定
主页
超级用户
日志
模块

- 开始安装，直到安装完成，点击 `重启`
  -


13:58
安装
完成!
- Copying zip to temp directory
- Installing Shamiko-v0.7.3-174-release.zip
- Current boot slot: _a
- Device is system-as-root
****
Shamiko
by LSPosed Developers
****
****
Powered by Magisk
****
- Shamiko version v0.7.3 (174)
- Extracting verify.sh
- Verified META-INF/com/google/android/update-binary
- Verified customize.sh
- Verified verify.sh
- Installing from Magisk / KernelSU app
- Magisk version: 26100
- Device platform: arm64
- Verified checksum
- Verified module.prop
- Verified sepolicy.rule
- Verified post-fs-data.sh
- Verified service.sh
- Verified uninstall.sh
- Extracting zygisk libraries
- Verified zygisk/armeabi-v7a.so
- Verified zygisk/arm64-v8a.so
- これで勝ったと思うなよ——!!
- Installing custom sepolicy rules
- Done
重启

- 重启Magisk后，即可看到：已安装的模块 `Shamiko`
  -


14:00
模块
从本地安装
Shamiko
v0.7.3 (174)，作者 LSPosed Developers
[😋 Shamiko is working as blacklist mode] これで勝ったと思うなよーーー!!
移除
Zygisk - LSPosed
v1.8.6 (6712)，作者 LSPosed Developers
Another enhanced implementation of Xposed Framework. Supports Android 8.1 ~ 13. Requires Magisk 24.0+ and Zygisk enabled.
移除
主页
超级用户
日志
模块

使用Xposed

安装了Xposed后，对于Xposed的使用，主要是：

- 安装各种 `插件`
- 使用各种 `插件`
  - 实现特定的目的和效果

注：Xposed的 `插件` = `plugin` = `模块` = `module`

Xposed插件

下面介绍一些常用的Xposed插件。

JustTrustMe

- `JustTrustMe`
  - 绕过HTTPS证书校验
- XPosed中安装JustTrustMe · 好用的安卓模拟器：夜神Nox (crifan.org)
- 【已解决】Mac中夜神模拟器中安装Xposed模块：JustTrustMe
  - 然后就可以去用Charles抓包了，而绕开https了，实现了破解https的目的
    - app抓包利器：Charles

砸壳导出dex

FDex2

FDex2 · Android逆向：静态分析

DumpDex

详见：

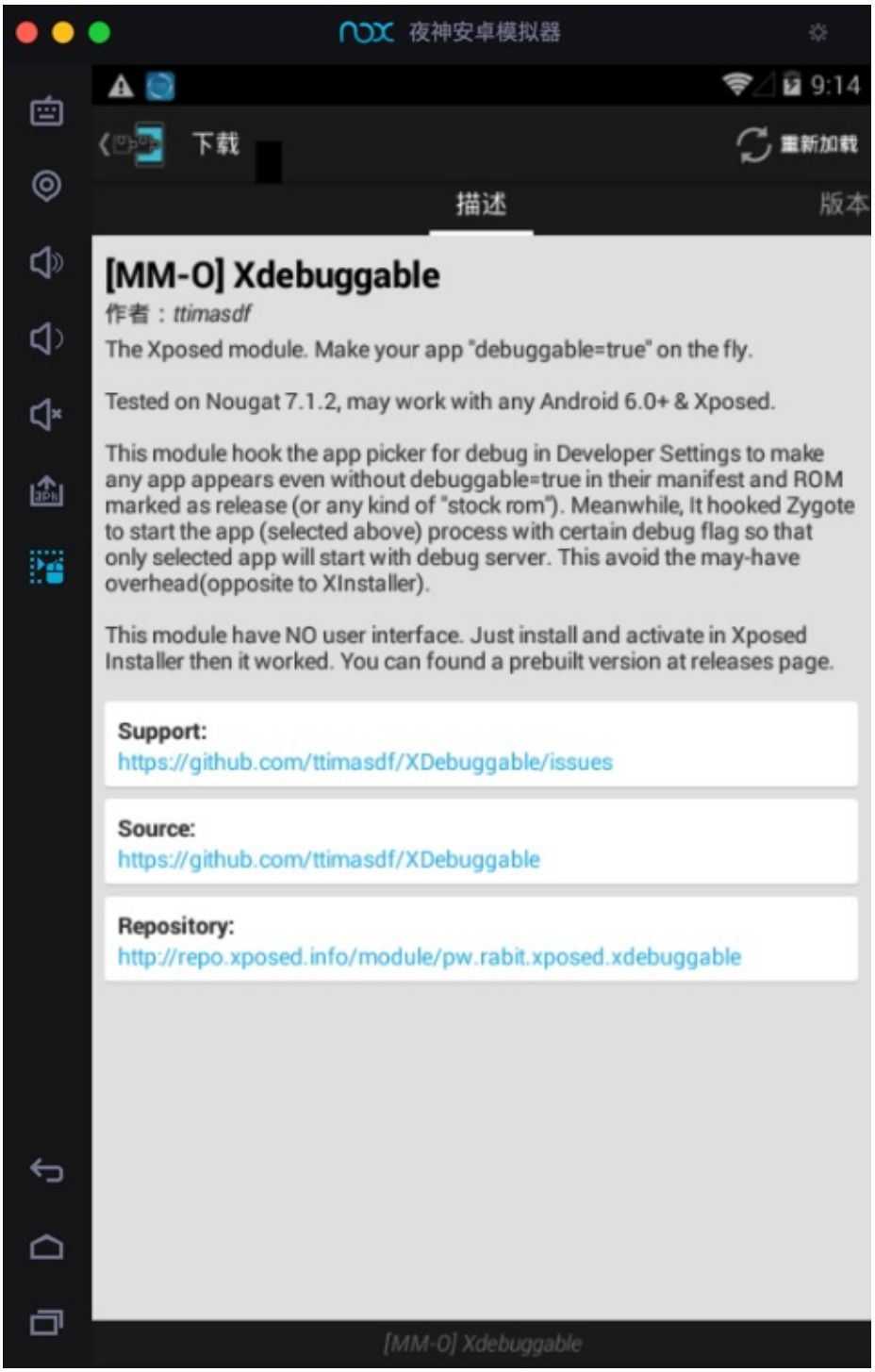
DumpDex · Android逆向：静态分析

具体用法：

配合XPosed插件导出dex · 好用的安卓模拟器：夜神Nox (crifan.org)

让app可调试的插件

`Xdebuggable`

- Xdebuggable
  - 旧页面：均已失效
    - [MM-O] Xdebuggable | Xposed Module Repository
      - 旧版Xposed可以搜到相关插件
      - 

      - apk下载地址：
        - https://dl-xda.xposed.info/modules/pw.rabit.xposed.xdebuggable_v4_c319ea.apk
  - 新来源

        - https://github.com/ttimasdf/XDebuggable
            - apk下载
                - https://github.com/ttimasdf/XDebuggable/releases
                    - 举例
                        - https://github.com/ttimasdf/XDebuggable/releases/download/v1.3/app-release.apk

XAppDebug

- XAppDebug
    - Github
        - https://github.com/Palatis/XAppDebug
            - 下载
                - https://github.com/Palatis/XAppDebug/releases
    - LSPosed插件页面
        - XAppDebug - Xposed Module Repository
            - https://modules.lsposed.org/module/tw.idv.palatis.xappdebug

GravityBox

- GravityBox
  - 是什么：Xposed插件
  - 作用
    - 各种功能
      - 从更改锁定屏幕、状态栏的颜色（及其透明度）、修改标准的最小和最大亮度级别、停用烦人的图标、建立扩展桌面模式等等
    - 其他比如
      - 更改赋予 Android 终端硬件按钮的功能
      - 修改状态栏的外观和显示的选项
      - 改变可以在手机或平板电脑屏幕上显示的亮度（这个是积极的，尤其是最小的，以节省电池）
      - 直接应用程序分配给设备的触摸按钮
      - 管理终端 RAM 的使用，了解此应用程序的消耗
      - 修改锁屏
      - 即使显示桌面也能旋转屏幕
    - 其他介绍
      - 安卓4.4电池图标美化、状态栏图标美化、时间居中显示、支持数字电量、全局透明度开启、隐藏SIM卡提示和显示、增加高级电源菜单、修改快捷下拉栏、简单开启虚拟按键功能、自带按键救星功能、关机菜单加入截屏功能、旧式电视CRT关屏特效、恢复开发者选项、锁屏快速程序启动
      - 重力工具箱GravityBox是一款依赖Xposed框架支持的全能系统工具，支持修改ROM的部分系统级重要参数
      - Xposed框架上必备神器级模块，软件是原生支持中文版本，可以修改和定制非常多系统级的功能。原生这个工具只能用于MTK平台，现在作者已经将它弄成通用的了
  - 截图
    -


3.9KB/s
9:24
电源调整
高级重新启动菜单
启用重新启动和Recovery选项高级菜单
电源键菜单屏幕截图
低电量警告
弹出窗口以及音效
停用LED闪烁
停掉当低电量且未充电时会出现扰人的LED闪烁
禁用充电LED灯
禁用当充电时的LED灯

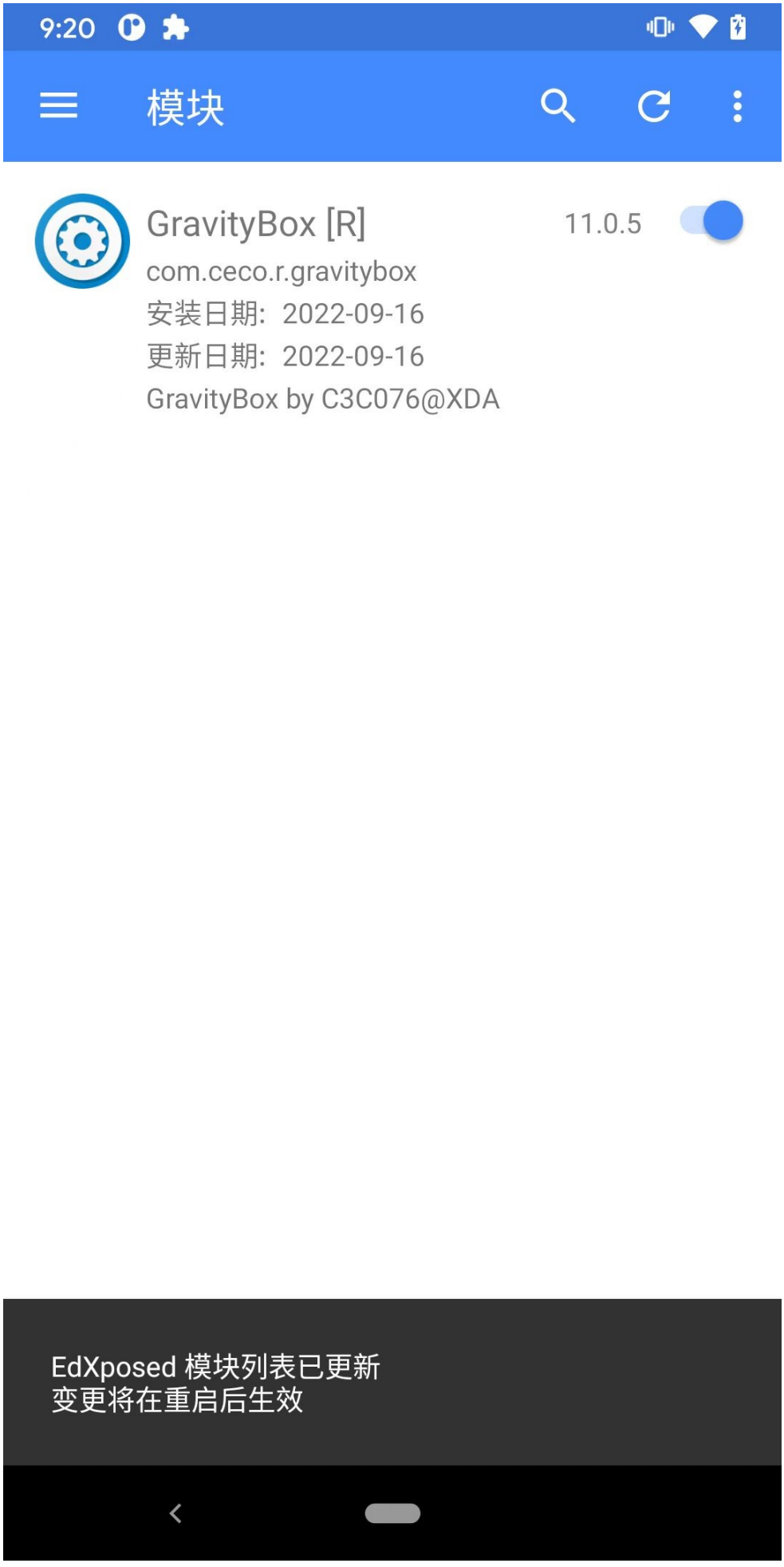

2.5KB/s
9:14
媒体调整
音量键跳过歌曲
当屏幕关闭时，长按音量键以跳过歌曲
更多音量段数
重新启动
重新启动
软重启
Recovery
Bootloader
取消
禁用时铃声音量与通知音量将可以分别调整
静音音量调节音效
禁用调节音量时发出音效

试用GravityBox

从Github的release页面下载到apk：

https://github.com/GravityBox/GravityBox/releases/download/v11.0.5_r/GravityBox_R_11.0.5.apk

然后安装，安装后可以在 `EdXposed Manager` 中看到Xposed插件，点击开启：


9:20
模块
GravityBox [R]
11.0.5
com.ceco.r.gravitybox
安装日期: 2022-09-16
更新日期: 2022-09-16
GravityBox by C3C076@XDA
EdXposed 模块列表已更新
变更将在重启后生效

更多选项：


9:20
模块
GravityBox [R]
11.0.5
com.ceco.r.gravitybox
安装日期: 2022-09-16
更新日期: 2022-09-16
GravityBox by C3C076@XDA
打开
设置作用域
在应用商店查看
应用信息
卸载

点击打开：


9:20
GravityBox [R]
GravityBox [R]
正在等待系统框架响应
GravityBox...

首次有警告弹框：


9:29
GravityBox [R]
锁屏界面调整
包含各种锁屏界面的调整选项
状态栏调整
包含各种状态栏的调整选项
警告
仅适用原生安卓或Google Play Edition设备使用!
使用其它品牌或者定制ROM像:Touchwiz, Sense, Xperia可能会导致意想不到的结果
如果需要继续, 那么您确认这是您自己的风险,并没有任何技术支持与期望。有关兼容性的详细信息, 请看XDA 论坛的支持页面
取消
我明白了
电话调整
包含各种通话的调整选项
媒体调整
包含各种媒体和声音系统的调整选项
实体按键调整
包含各种与实体按键自定义动作相关的调整选项

主界面可以看到各种功能：


9:29
GravityBox [R]
锁屏界面调整
包含各种锁屏界面的调整选项
状态栏调整
包含各种状态栏的调整选项
虚拟按键调整
包含各种与虚拟按键相关的调整选项
Pie Controls虚拟按键调整
包含与Pie Controls控制相关的调整选项
电源键调整
包含各种与电源键相关的调整选项
显示调整
包含各种与显示相关的调整选项
电话调整
包含各种通话的调整选项
媒体调整
包含各种媒体和声音系统的调整选项
实体按键调整
包含各种与实体按键自定义动作相关的调整选项


9:29
GravityBox [R]
媒体调整
包含各种媒体和声音系统的调整选项
实体按键调整
包含各种与实体按键自定义动作相关的调整选项
应用程序启动器
对话框允许定义最多 12 个应用程序,可以分配实体按键或作为额外的虚拟按键
指纹启动器
允许在屏幕打开且设备未锁定时使用指纹传感器启动应用或操作
屏幕录制
包含各种与屏幕录制相关的调整选项
杂项调整
包含所有不属于特定类别的调整选项
终极通知控制
允许控制每个应用的通知指示灯、提示音和振动提示
高级调谐器
允许对框架和系统 UI 配置参数进行高级调整
关于

然后记录点相关功能：

- 实体按键调整
  -


9:30
实体按键调整
菜单键
单击动作
默认
长按动作
默认
双击动作
默认
主页键
长按动作
默认
禁用双击
双击动作
默认
返回键
单击动作
默认
长按动作
默认

- 电源键调整
  -


9:31
电源键调整
电源键菜单
在锁屏上禁用电源键菜单
高级重新启动菜单
启用重新启动和Recovery选项高级菜单
允许软重启
注意,软重启会引起 意想不到的电池消耗或不稳定问题,最好的方法是使用正常重启
允许在锁屏显示
重启需要确认
在电源键菜单显示扩展桌面按钮
其它
低电量警告
默认
电池充满声音
没有声音
插入充电器声音

Xposed插件开发

TODO：

- 【未解决】用FDex2核心源码开发出可以正常安装的安卓apk的XPosed插件
- 【记录】参考和翻译XPosed的官网关于如何开发XPosed插件的文档
- 【记录】确保Android Studio能用Nox安卓模拟器设备去调试
-
- 【已解决】Android Studio中实现一个空功能的基本的安卓apk
- 【未解决】给安卓应用中新增一个按钮和点击按钮改变文字
- 【已解决】安卓app开发：新增一个按钮并设置文字和点击响应事件函数
-
- 【记录】Nox夜神模拟器中查看已安装的Xposed插件的安卓应用
- 【已解决】把XPosed插件的安卓apk安装到Nox夜神安卓模拟器中
- 【记录】确认XPosed插件生效正常工作输入期望log日志信息
- 【已解决】研究XPosed Installer的log日志
- 【记录】给XPosed插件版本号增加为1.1且用AS重新调试安装
- 【已解决】XPosed插件报错：java.lang.IllegalAccessError Class ref in pre-verified class resolved to unexpected implementation
- 【未解决】Android Studio中如何调试XPosed插件
- 【无需解决】把XPosed插件的安卓apk安装到普通安卓AVD模拟器中
- 【未解决】Android Studio调试Xposed插件报错：Waiting for application to come online
- 【未解决】研究安卓设备的设置中的选择调试应用和等待调试器的含义和用法
- 【未解决】AS中调试外部app显示所有进程看不到我们的Xposed插件的app的进程
- 【未解决】Mac中其他好用且能安装Xposed框架的安卓模拟器
-
- 【记录】下载和研究Xposed的XposedTools源码
- 【未解决】Mac中自己去编译x86_64的sdk23的Xposed框架zip安装包
- 【未解决】换XPosed插件开发所需的安卓模拟器从Nox夜神换为网易Mumu

此处环境：

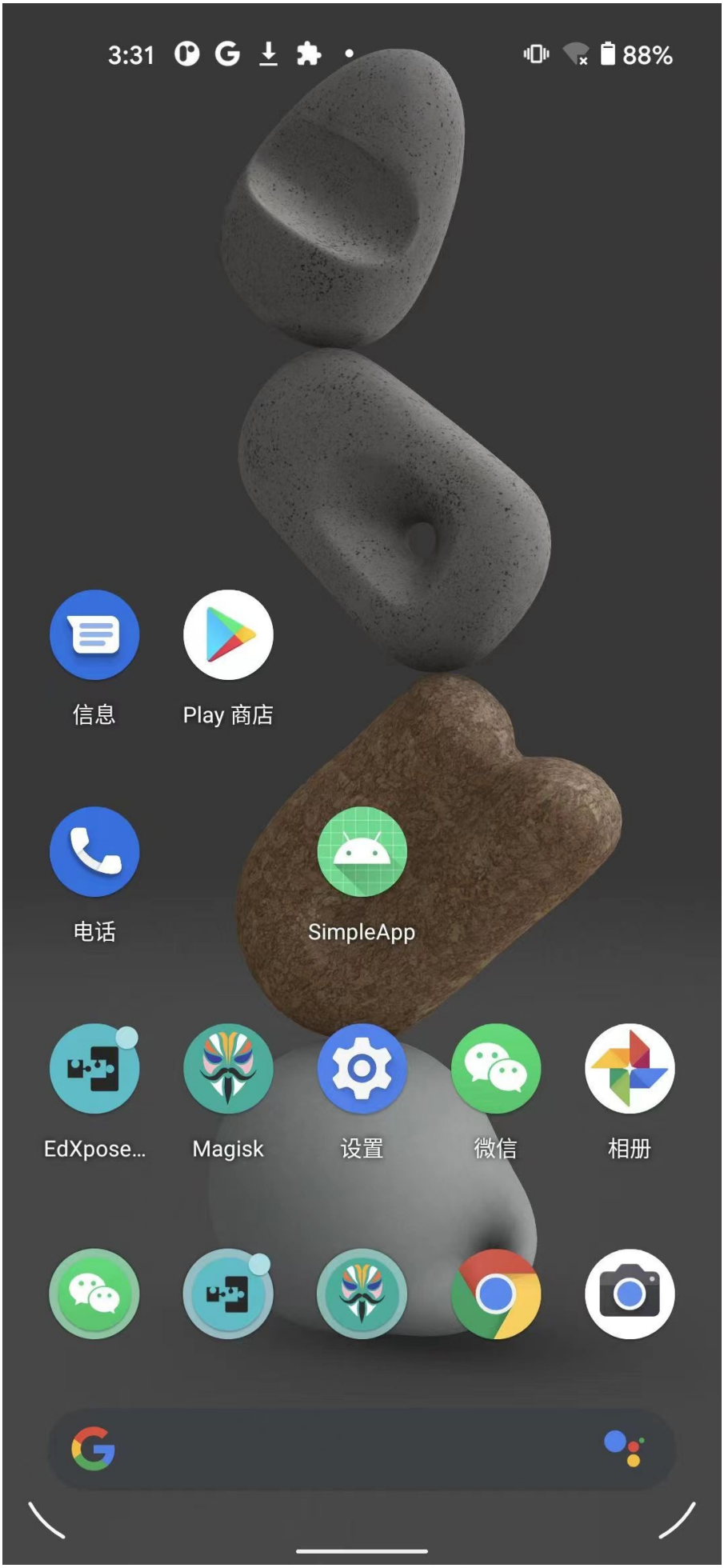
- 电脑：`Mac`
  - 已安装 `Android Studio`
  - 可用adb查看到安卓设备

```
➜  ~ adb devices -l
List of devices attached
9C181A8D3C3F3B         device usb:1048576X product:redfin model:Pixel_5 device:redfin transport_id:3
```

- 安卓手机：`Google Pixel 5`
  - `Android 11`
    - ■


5:08
83%
关于手机
设备名称
Pixel 5
手机号码
未知
急救信息
机主的相关信息和联系人信息
法律信息
监管标签
安全和监管手册
SIM 卡状态
无法获取
型号
Pixel 5
IMEI
352493102553519
Android 版本
11
IP 地址
fe80::dce9:ffff:fe2f:2f6e
192.168.2.17

- 已root，已安装：`EdXposed`、`Magisk`
  -


3:31
88%
信息
Play 商店
电话
SimpleApp
EdXpose...
Magisk
设置
微信
相册

下面介绍，如何开发一个最基本的**Xposed插件**：

- 用AndroidStudio创建基本的安卓app项目
- 改动项目，增加Xposed相关配置
  - 其中包括写Xposed插件代码
- 编译并安装Xposed的插件apk到安卓手机
- 确认Xposed插件的apk已安装，并测试插件是否生效

用AndroidStudio创建基本的安卓app项目

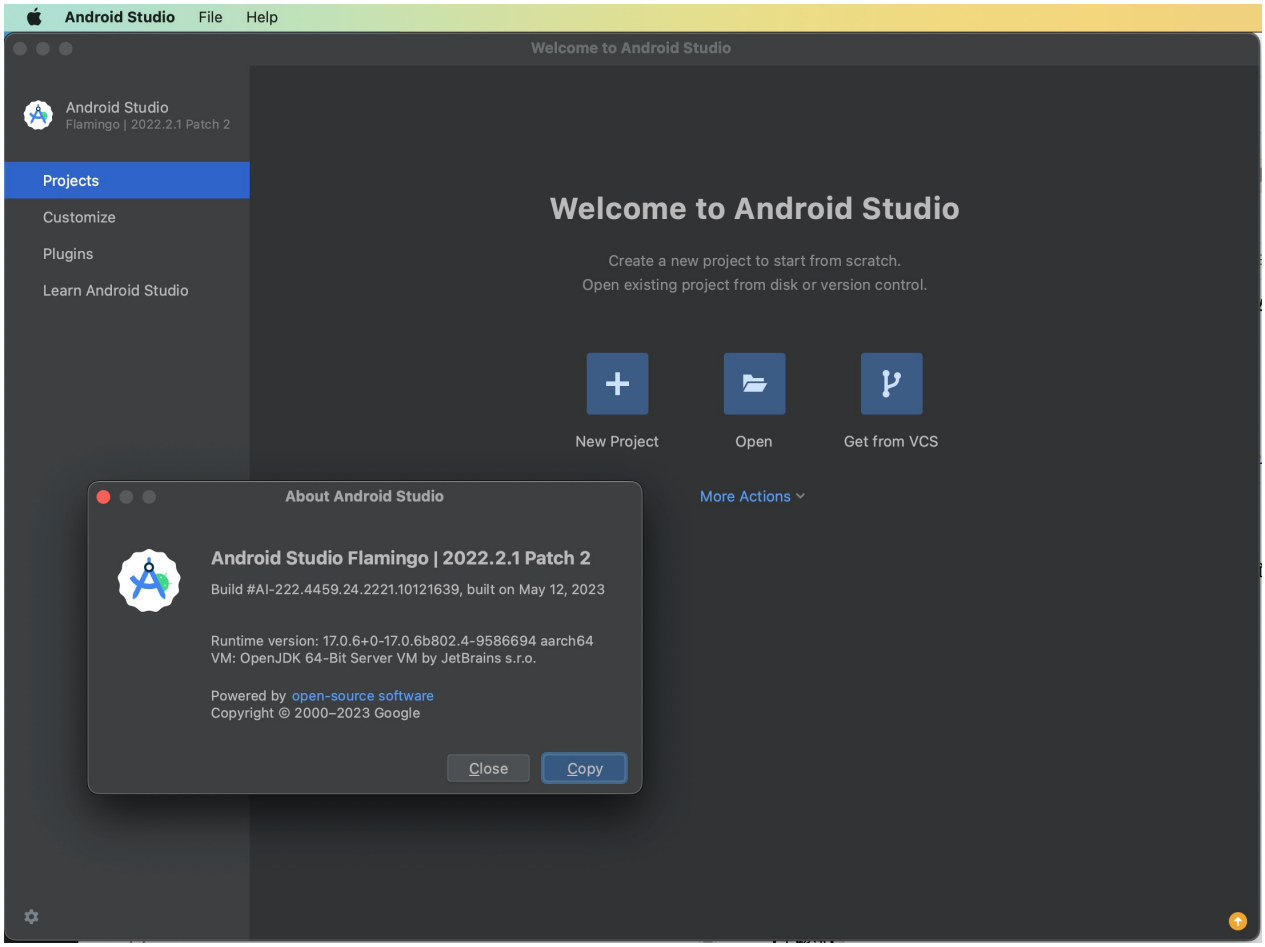
先用 `Android Studio` 去创建一个最基本的，普通的安卓app的项目：

- 核心步骤： `Android Studio` -> `New Project` -> `No Activity` ->设置项目基本属性： `Name` 、 `Package Name` 、 `Save Location` 、 `Language` 、 `Minimum SDK` -> `Finish`

详细步骤：

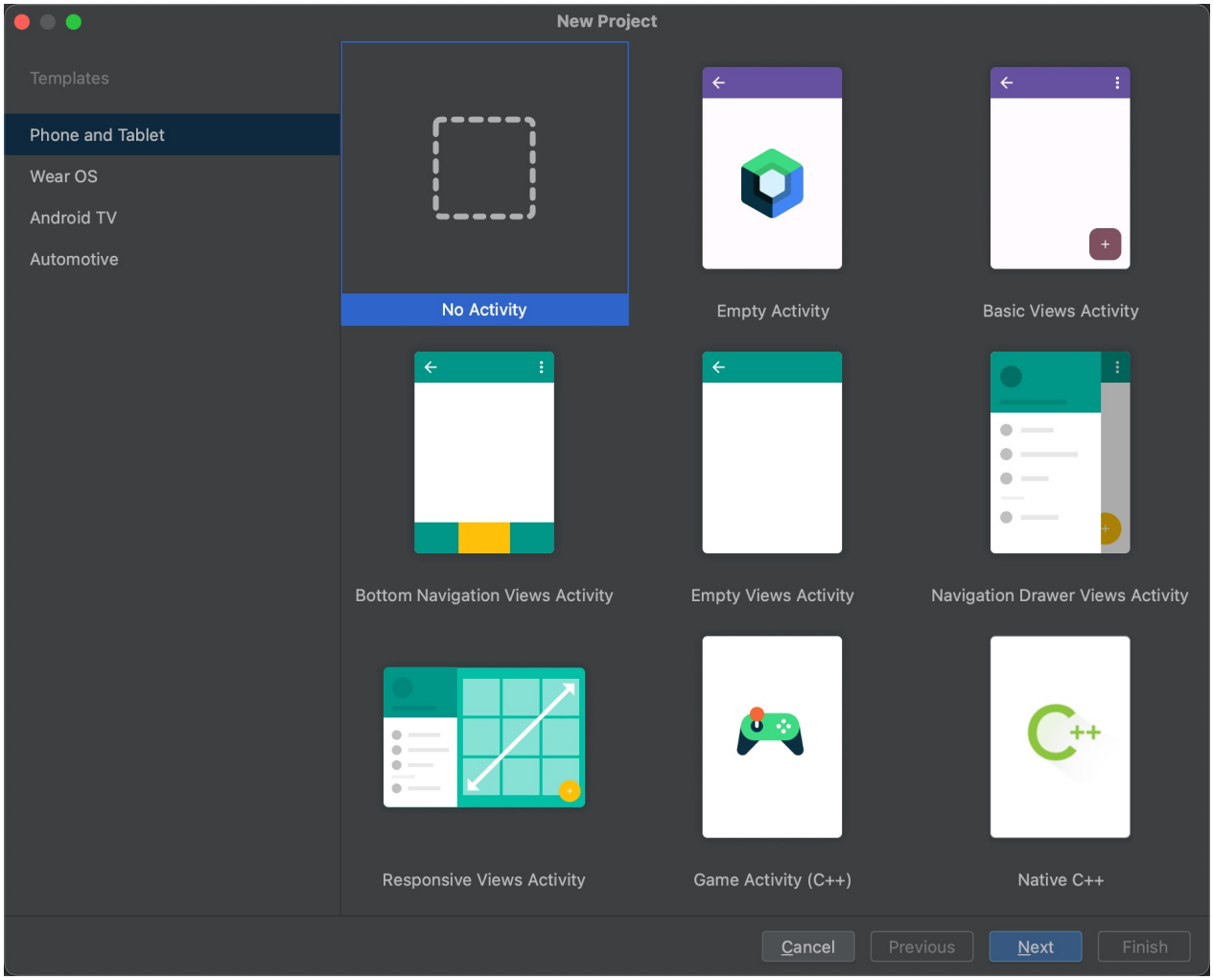
新建项目

点击： `New Project`



选择类型：No Activity

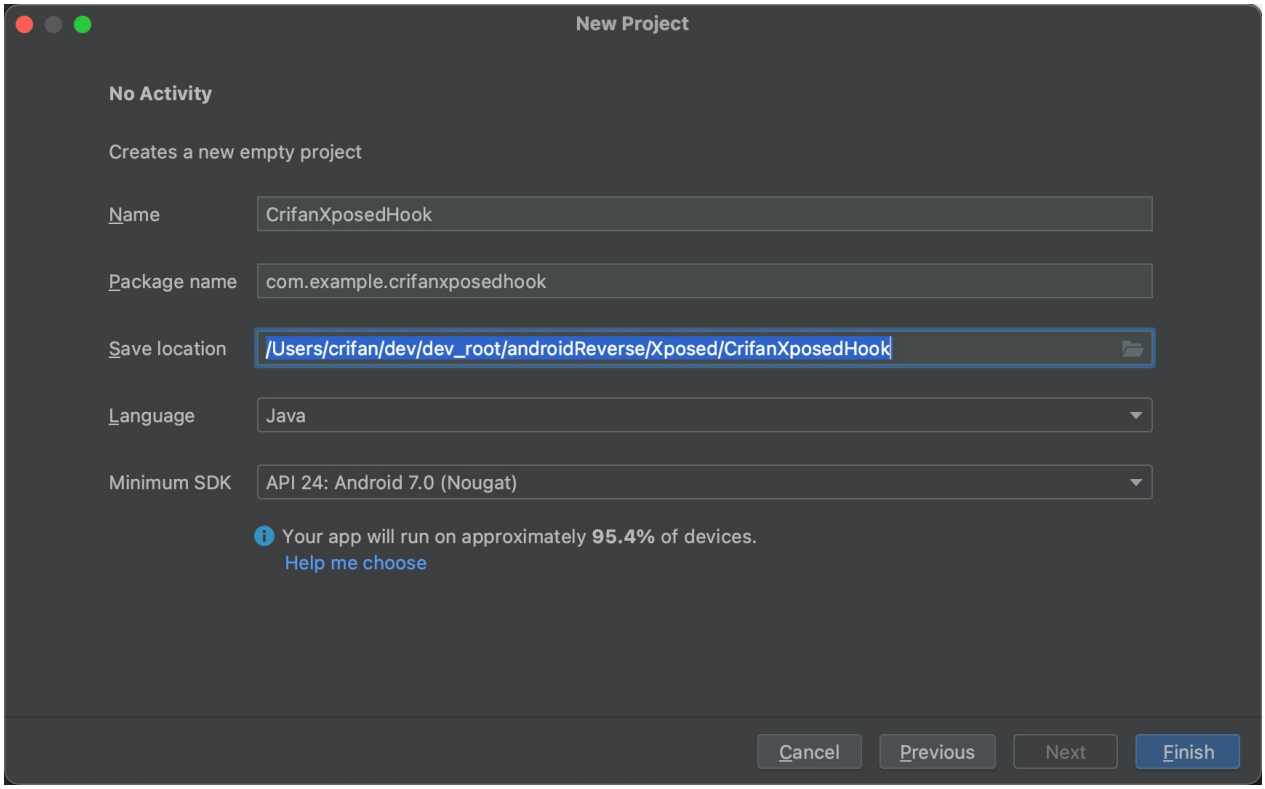
此处选择： `No Activity` =无页面

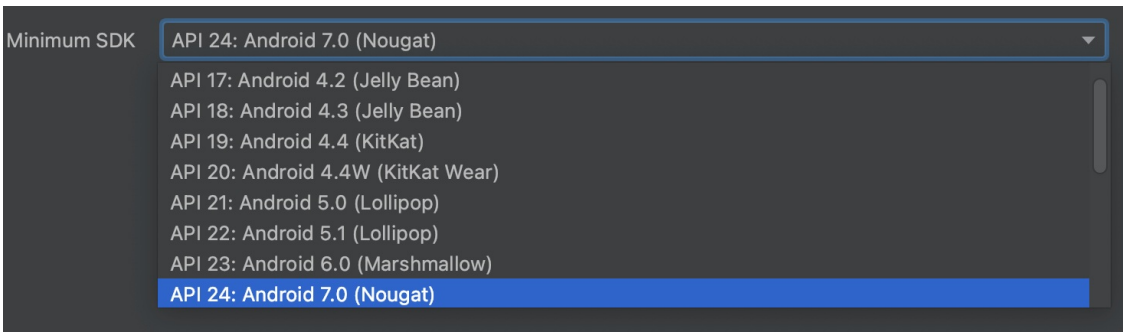


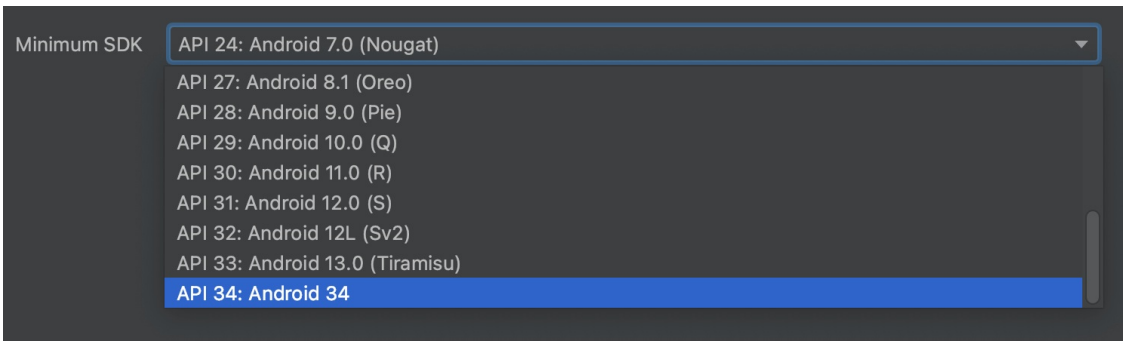
- 说明
  - 因为此处最基本的Xposed插件，无需页面
    - 只要触发了对应的hook代码，能输出log，就表示真正生效了
  - 另外：复杂的Xposed插件，可以根据自己需要去加上页面
    - 用以支持各种配置和逻辑等等更多其他内容

设置项目基本参数

继续去设置安卓项目的基本参数：



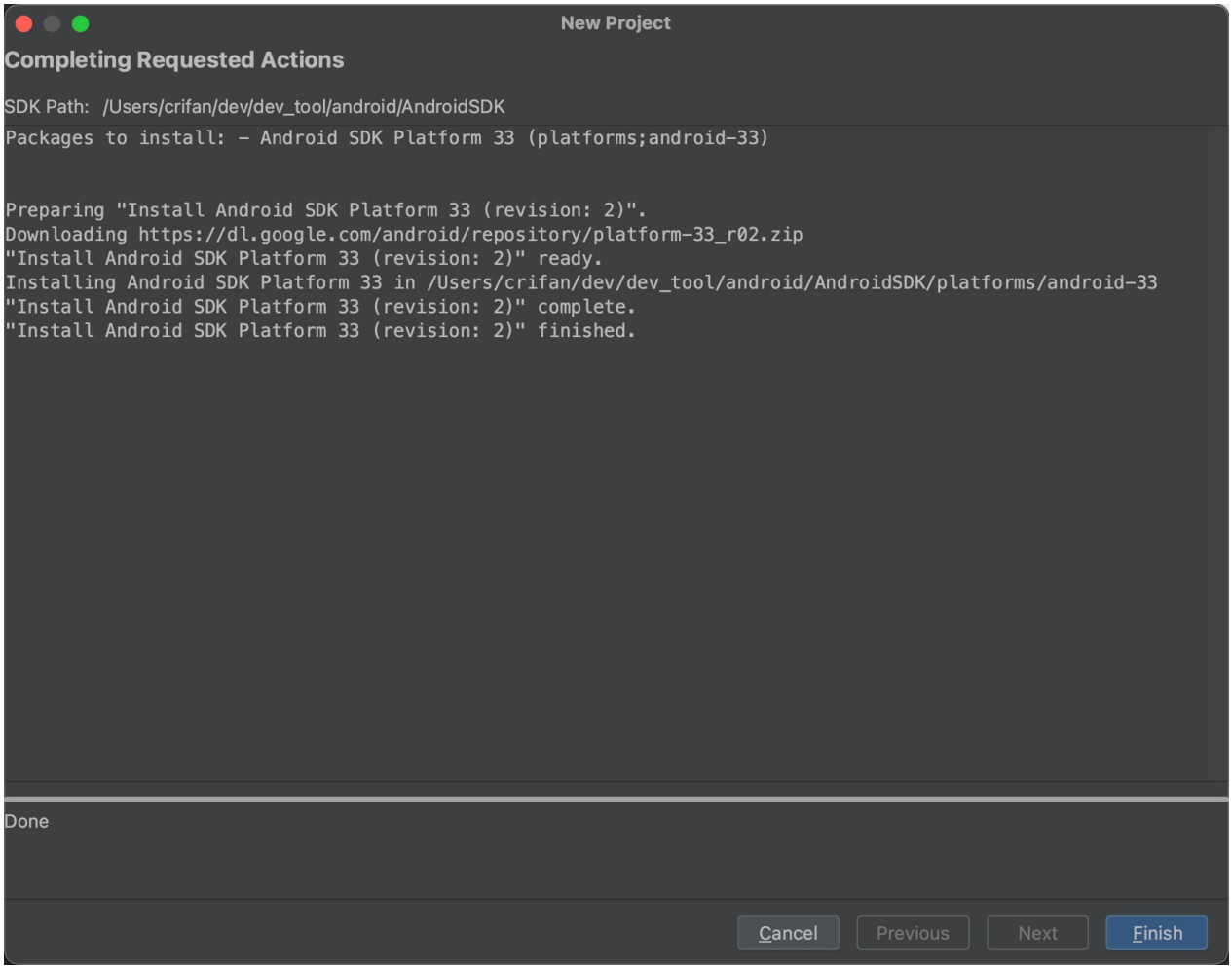
- Name：`CrifanXposedHook`
- Package Name：`com.example.crifanxposedhook`
- Save Location：`/Users/crifan/dev/dev_root/androidReverse/Xposed/CrifanXposedHook`
- Language：`Java`
  - 为了保险起见，先选`Java`，等熟悉了，再换到`Kotlin`
- Minimum SDK：`API 24: Android 7.0 (Nougat)`
  - 其他选项
    - 

    - 

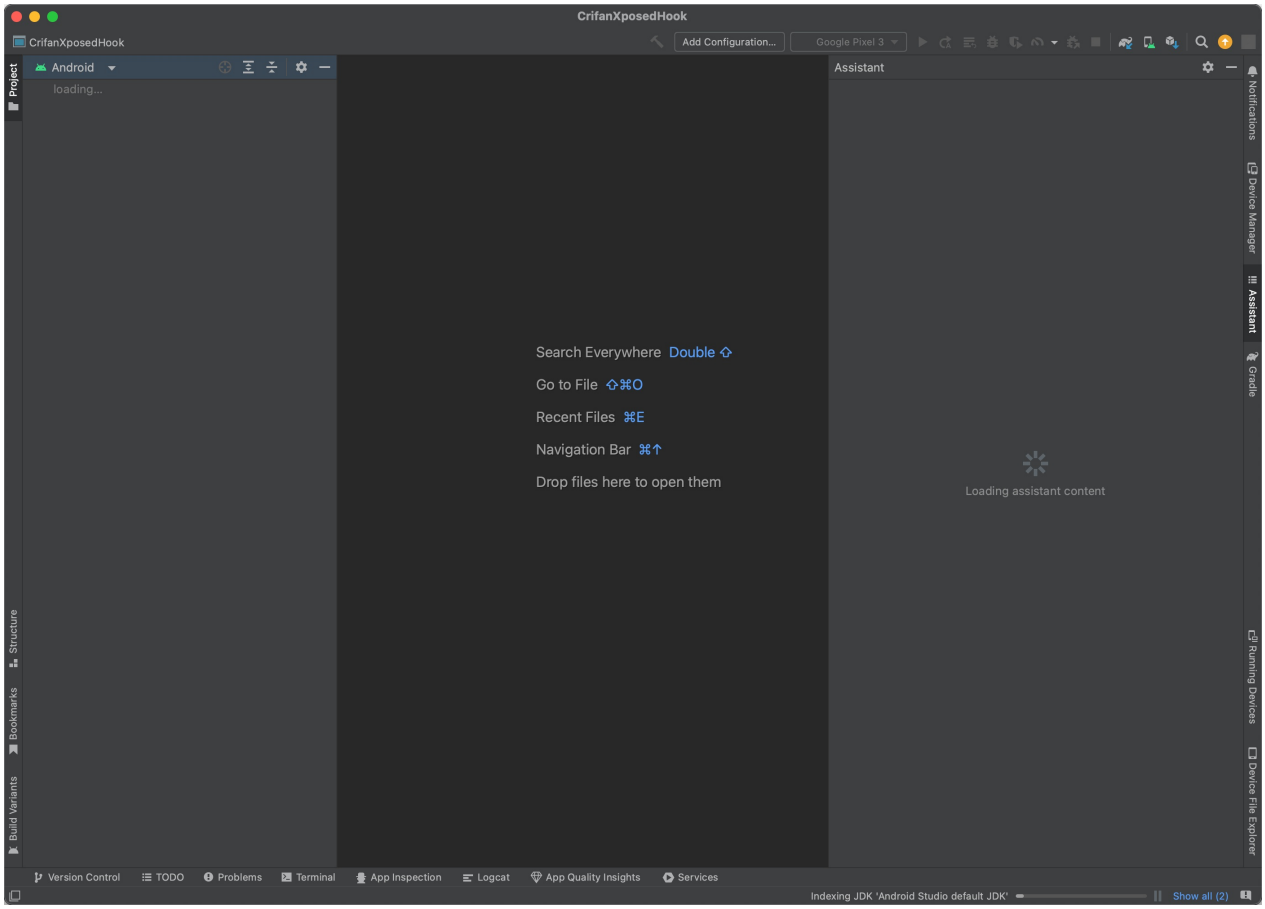
  - 可以根据自己需要，改为其他版本，比如更高版本的：`API 27: Android 8.1 (Oreo)`

初始化项目

点击`Finish`后，即可开始项目初始化，下载相关SDK等内容，直到下载完成：

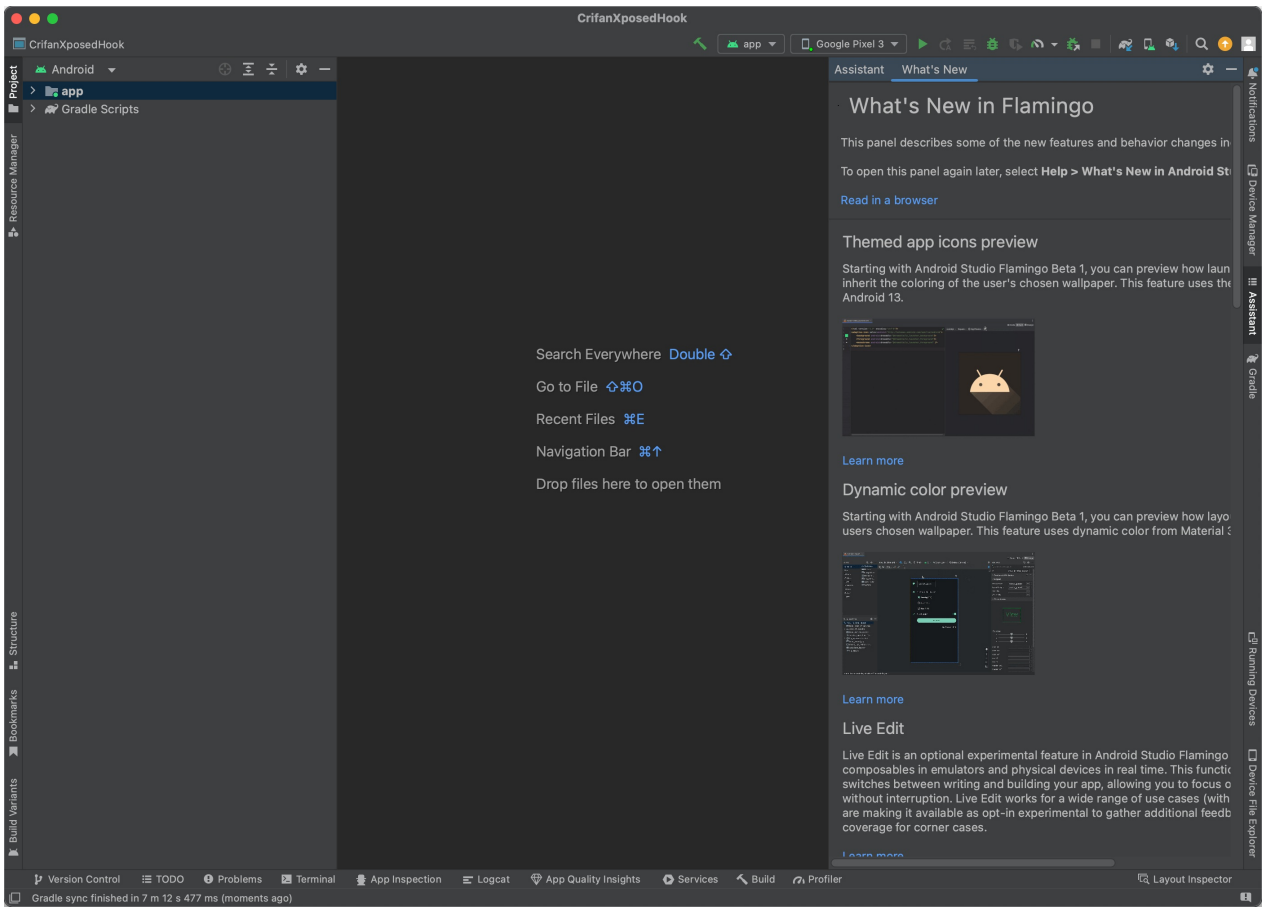


进入Android Studio的主页面，开始项目初始化：



后续会继续初始化，包括Gradle等相关资源的下载、导入、重建索引等。

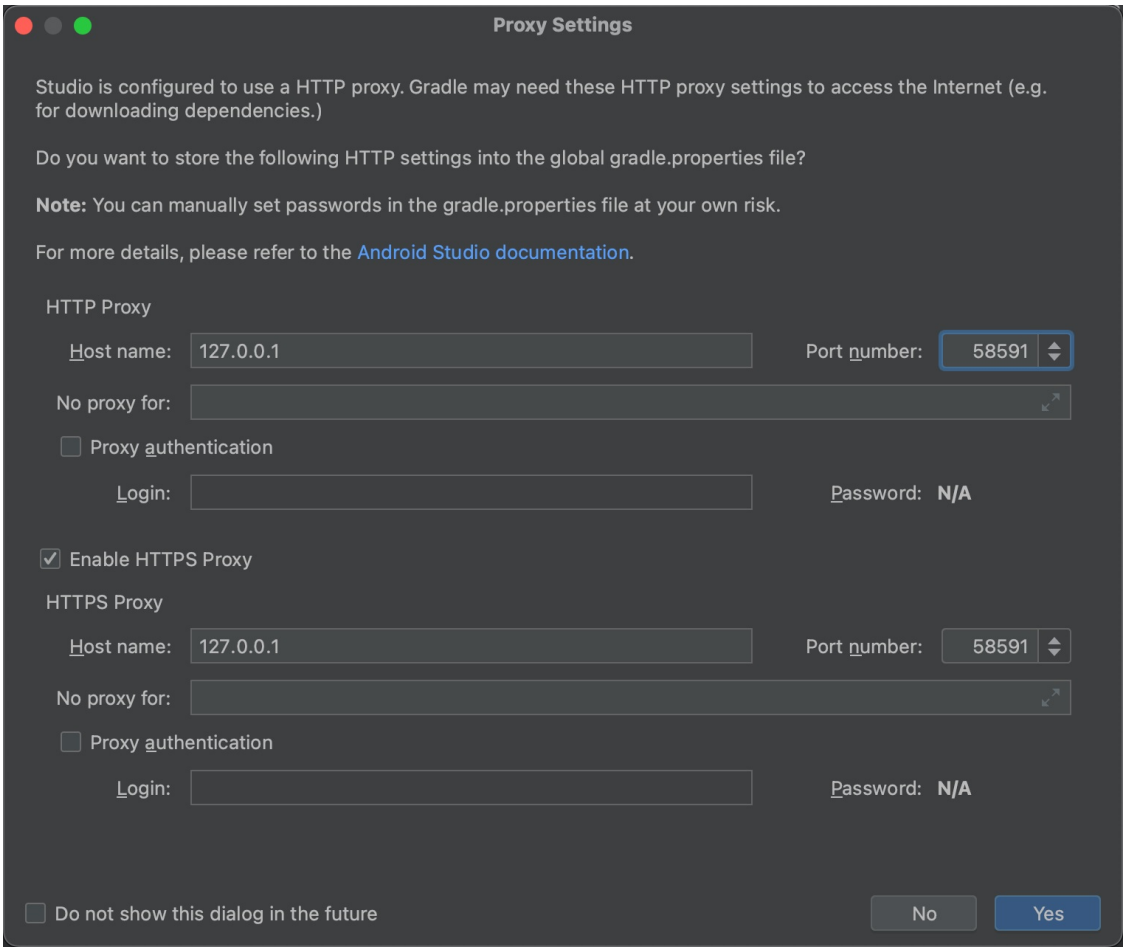
最后初始化完毕：



设置代理

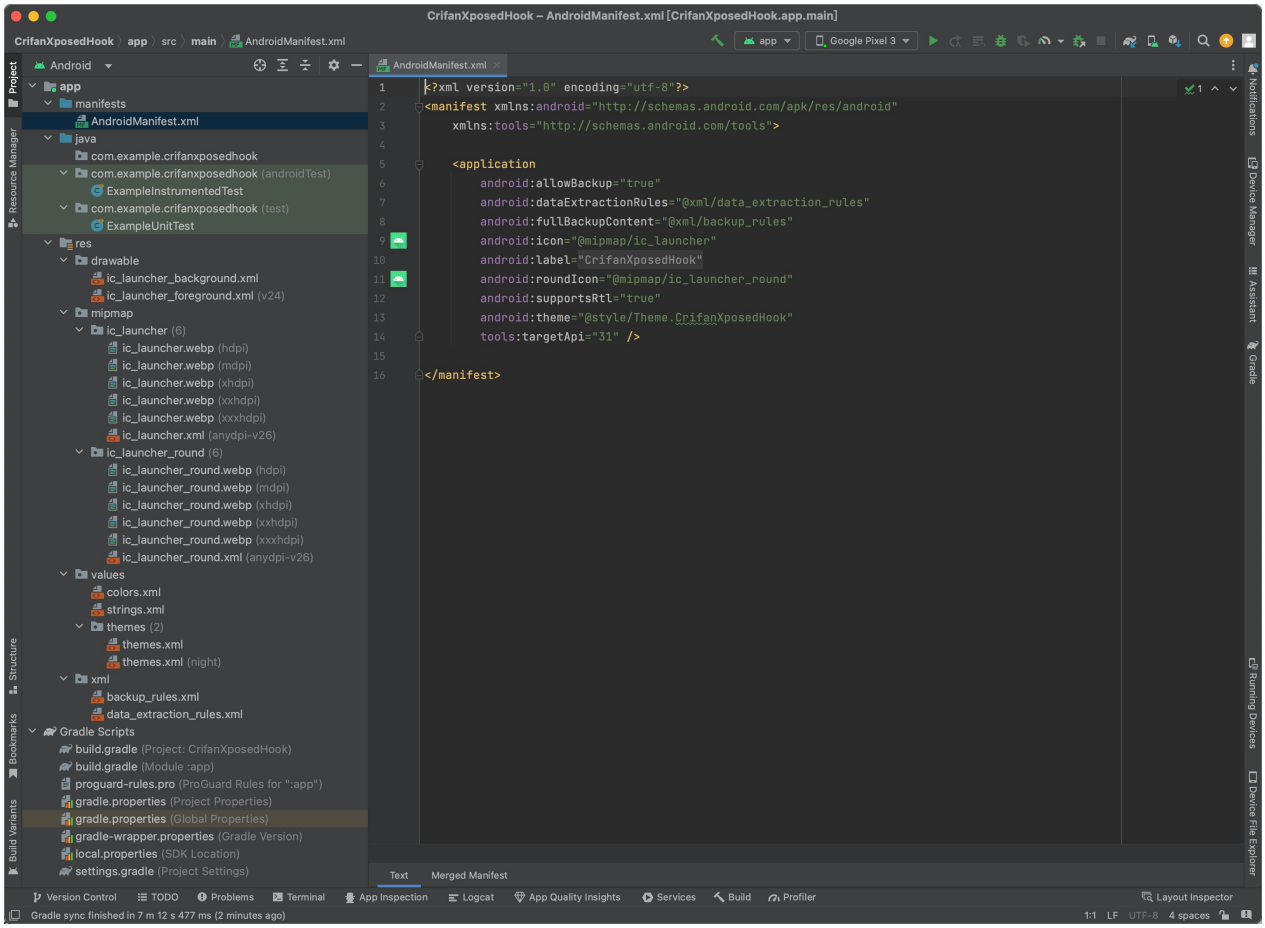
- 由于，从google等国外网站下载Google开发相关的资源，比如SDK等，国内网络经常无法连接或下载速度慢，所以此处往往还会弹框，问你是否要加代理
  - 此处选择添加代理，使用此处Mac中已有的Trojan的代理：

```
export HTTP_PROXY http://127.0.0.1:58591; export HTTPS_PROXY http://127.0.0.1:58591; export ALL_PROXY socks5://127.0.0.1:51837
```

  - 对应代理设置是：
    - 

最终项目文件和结构

点击展开目录结构如下：



改动项目，增加Xposed相关配置

新增改动或文件

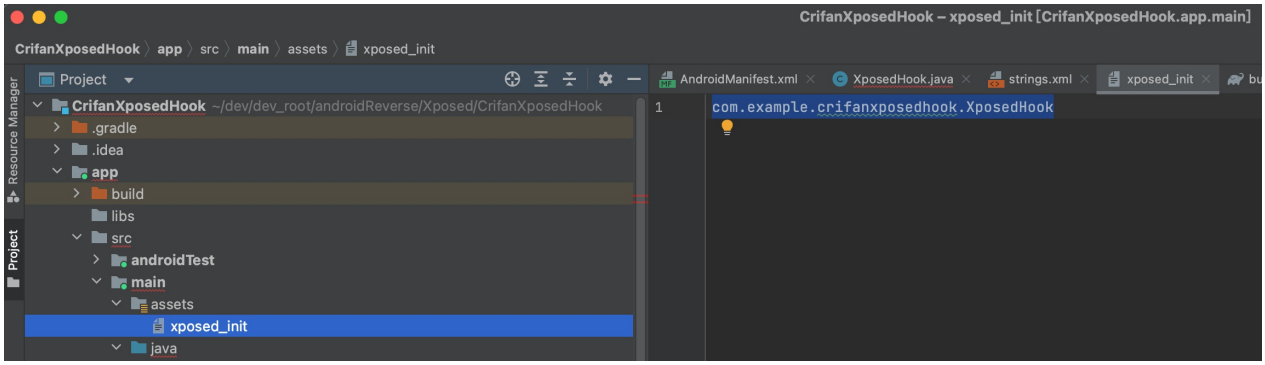
新建xposed_init

新建文件：`app/src/main/assets/xposed_init`

目的：定义插件hook代码所在的类

文件内容：

```
com.example.crifanxposedhook.XposedHook
```



内容说明：

- `com.example.crifanxposedhook.XposedHook`
  - 是后续的具体的hook插件代码对应的Java的类

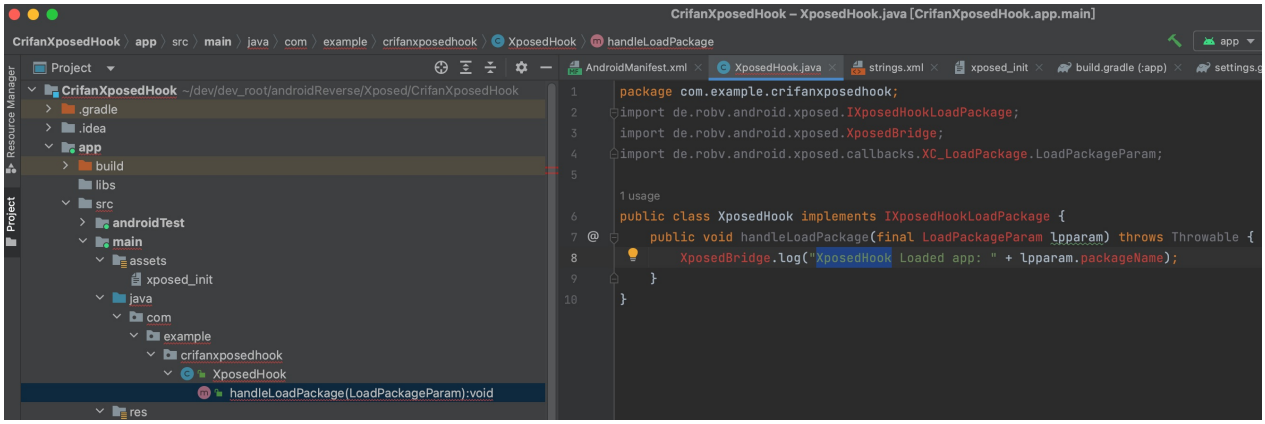
新建hook的Java类文件：XposedHook.java

新建hook代码文件：`app/src/main/java/com/example/crifanxposedhook/XposedHook.java`

目的： 实现hook代码逻辑

文件内容：

```
package                          ;
import                      IXposedHookLoadPackage;
import                      XposedBridge;
import                           XC_LoadPackage.LoadPackageParam;

public class XposedHook implements IXposedHookLoadPackage {
    public void handleLoadPackage(final LoadPackageParam lpparam) throws Throwable {
        XposedBridge.log("XposedHook Loaded app: " + lpparam.packageName);
    }
}
```



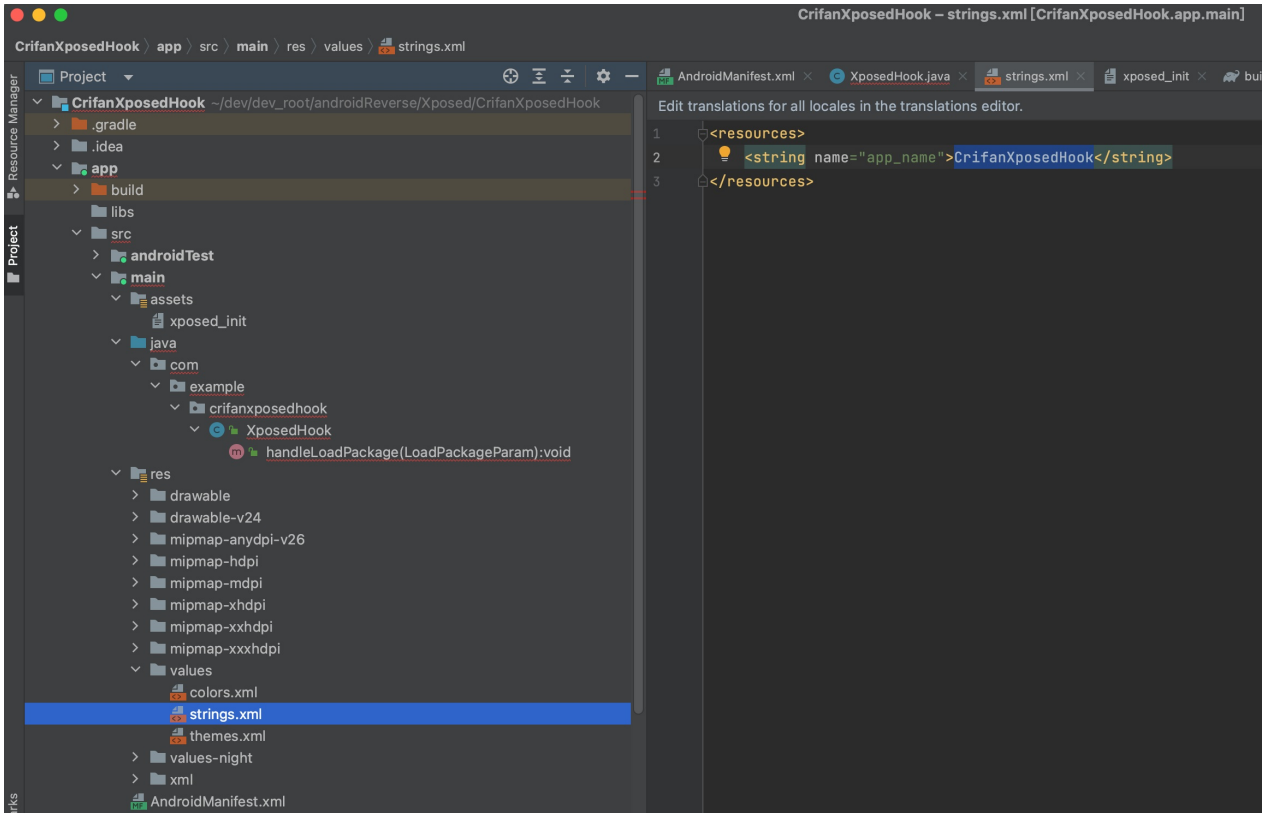
- 代码说明
  - 无需理会报错：`Cannot resolve symbol IXposedHookLoadPackage`
    - 基本上确定是`Android Studio`的bug?
  - 如果后续忘了加上`jcenter()`，则此处会报错找不到类：`Cannot resolve symbol de`
    - 需要记得加上`jcenter()`，详见后续内容

本身已有app名称定义：strings.xml

本身已有：`app/src/main/res/values/strings.xml`

文件内容：

```
<resources>
    <string name="app_name">CrifanXposedHook</string>
</resources>
```



- 内容说明
  - 其中`CrifanXposedHook`是当前app名称=Xposed插件名称=安卓项目名称

给AndroidManifest.xml中加上xposed相关定义

改动：`app/src/main/AndroidManifest.xml`

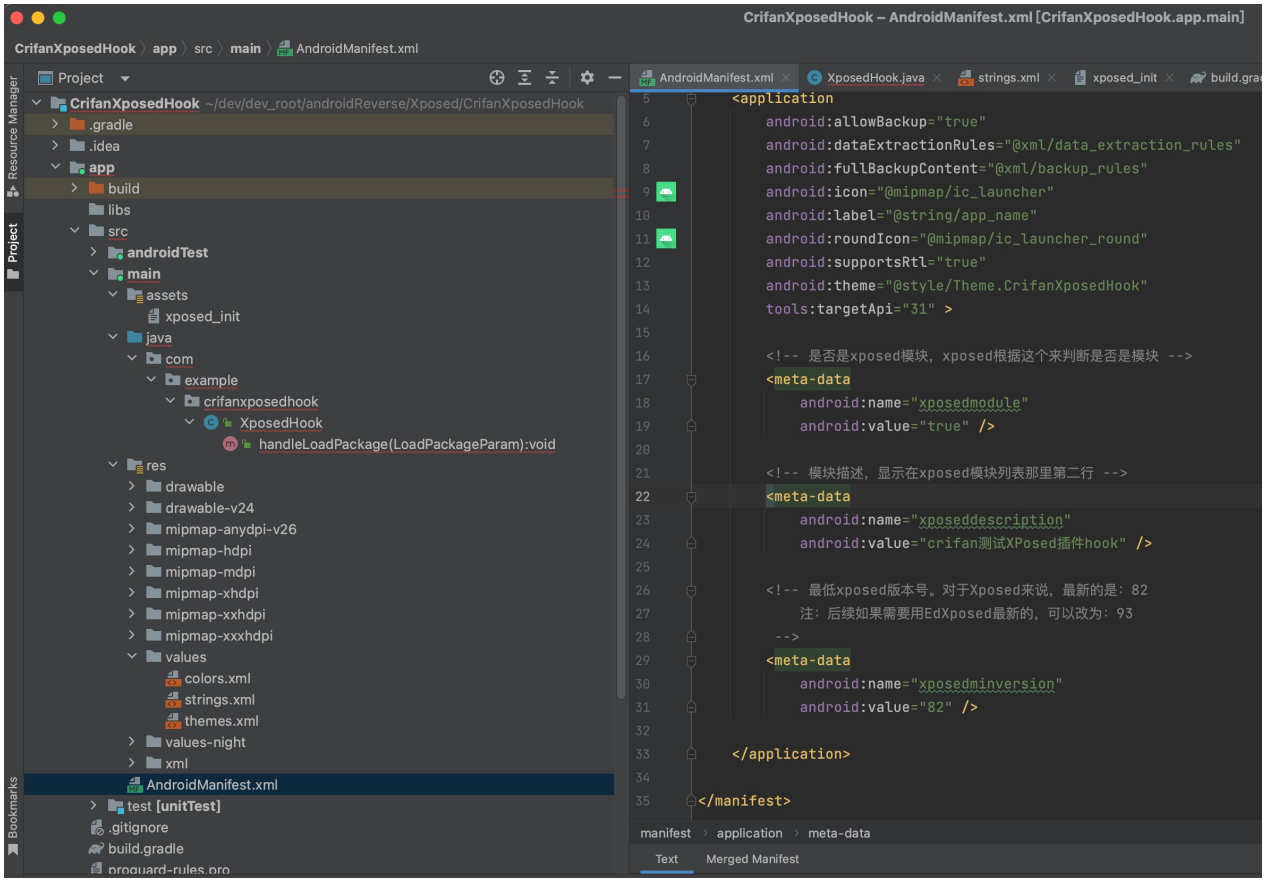
目的：新增Xposed相关配置 -> 让EdXposed等Xposed框架可以识别此（安卓的app是）Xposed插件

核心改动：修改 `application` 的部分属性，并加上 `meta-data` 的 `Xposed` 相关属性

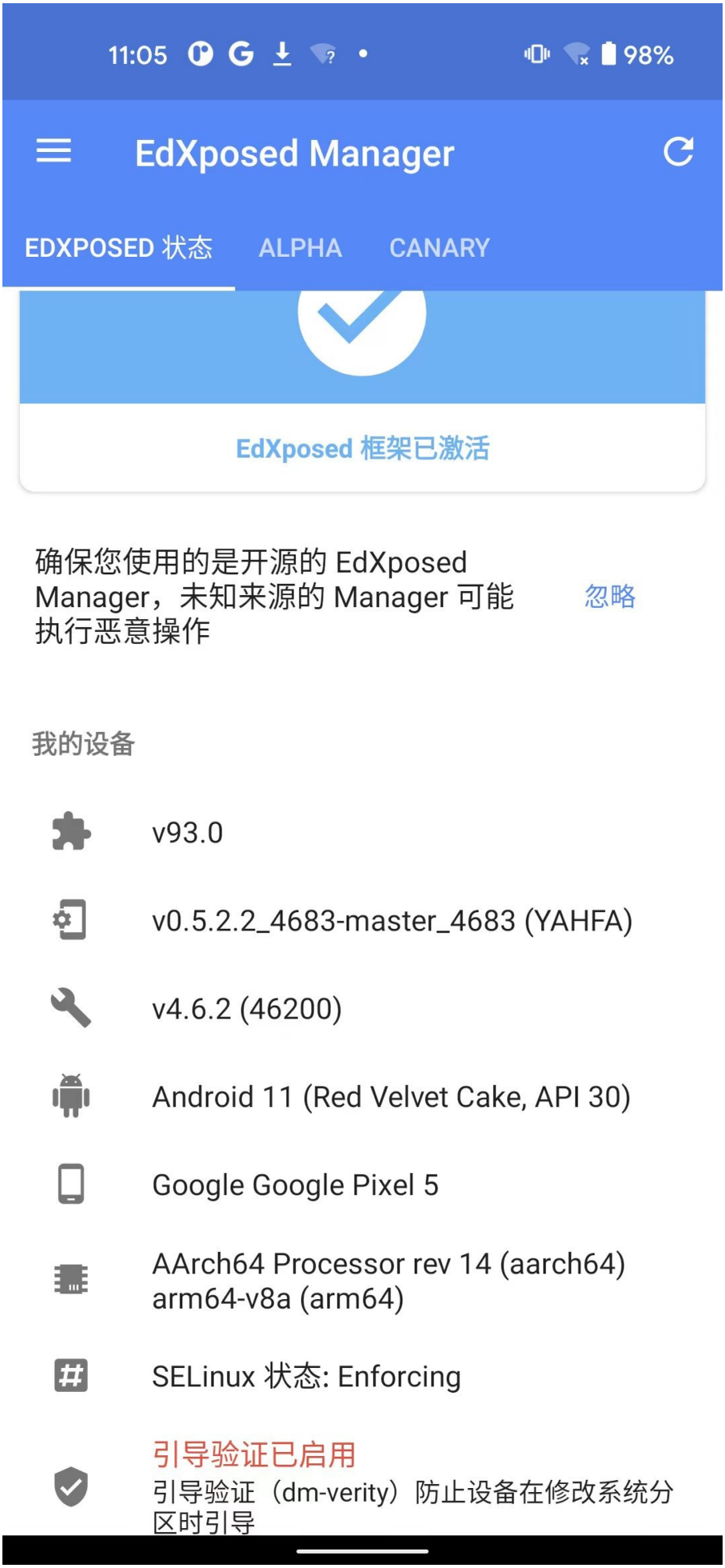
改动后的：

文件内容：

```
<?xml version="1.0" encoding="utf-8"?>
<manifest       android="http://schemas.android.com/apk/res/android"
          tools="http://schemas.android.com/tools">

    <application
                allowBackup="true"
                dataExtractionRules="@xml/data_extraction_rules"
                fullBackupContent="@xml/backup_rules"
                icon="@mipmap/ic_launcher"
                label="@string/app_name"
                roundIcon="@mipmap/ic_launcher_round"
                supportsRtl="true"
                theme="@style/Theme.CrifanXposedHook"
              targetApi="31" >

        <!-- 是否是xposed模块，xposed根据这个来判断是否是模块 -->
        <meta-data
                    name="xposedmodule"
                    value="true" />

        <!-- 模块描述，显示在xposed模块列表那里第二行 -->
        <meta-data
                    name="xposeddescription"
                    value="crifan测试XPosed插件hook" />

        <!-- 最低xposed版本号。对于Xposed来说，最新的是：82
            注：后续如果需要用EdXposed最新的，可以改为：93
         -->
        <meta-data
                    name="xposedminversion"
                    value="82" />
    </application>

</manifest>
```



- 参数说明
  - `xposedmodule` = `true`
    - 表示是Xposed插件=Xposed模块
  - `xposeddescription` = `crifan测试XPosed插件hook`
    - 插件描述，会显示=出现在Xposed模块列表中的文字描述部分
    - `xposedminversion` = `82`
    - 表示：`Xposed Framework API` = `Xposed框架的API版本`
      - 此处值：82
        - 最新的Xposed框架的API版本就是82
          - 不过其实也是好多年前了（好像是2016年，当前此刻是2023年）
        - 对应支持Android版本：`> Android 5.0`
          - 对于后续要测试的安卓手机的安卓版本（`Android 8.1`，`Android 11`，`Android 13`）等，完全满足此要求
      - 可选值=所有版本：`36`、`37`、`39`、`42`、`50`、`51`、`52`、`53`、`60`、`63`、`65`、`81`、`82`
        - 详见：IXposedHookLoadPackage | Xposed Framework API
      - 另外最好：去你的安卓手机中确认当前Xposed的API版本
        - 比如
          - 此处的Pixel5中，EdXposed首页中的：v93.0
            -

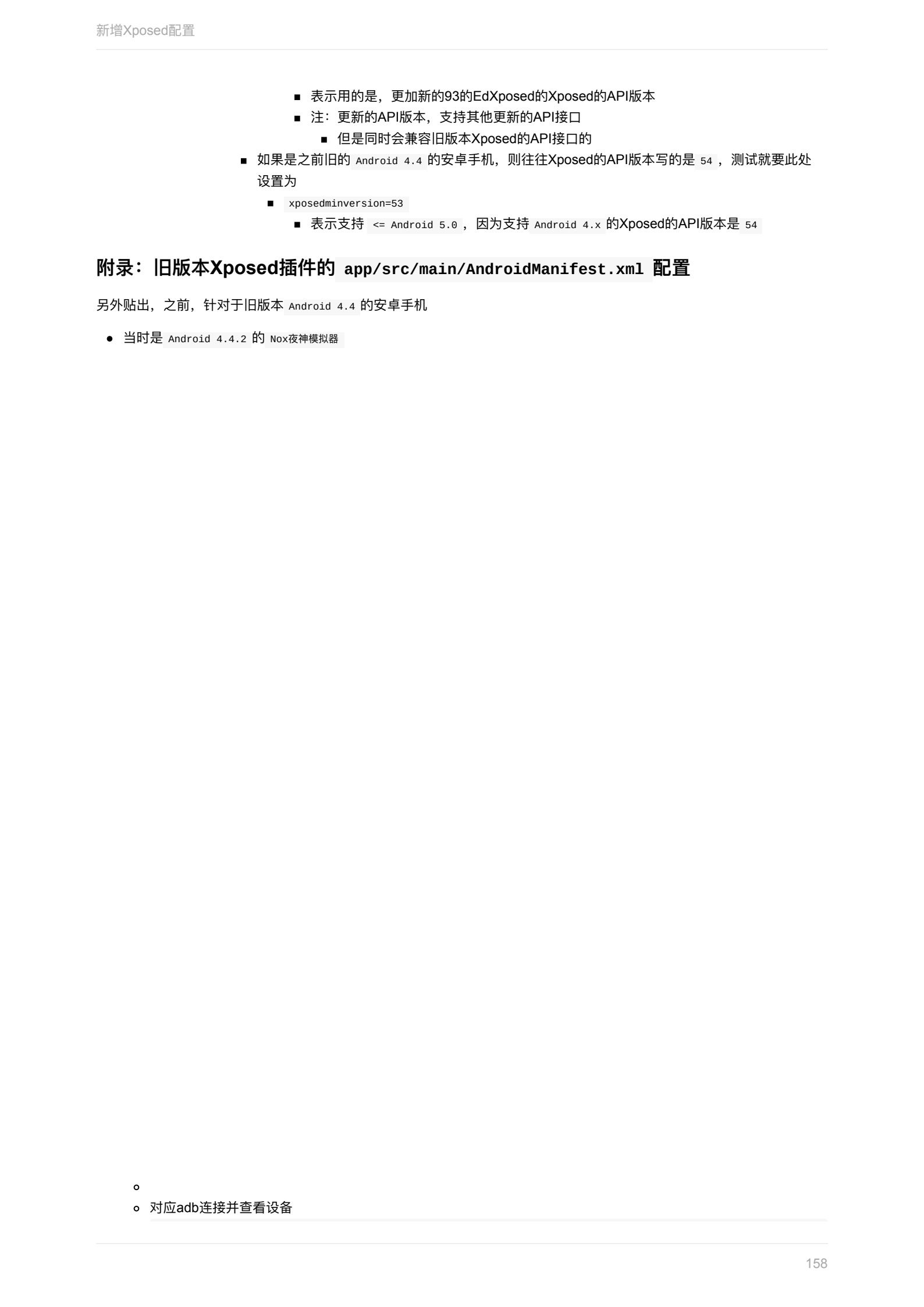

11:05
98%
EdXposed Manager
EDXPOSED 状态
ALPHA
CANARY
EdXposed 框架已激活
确保您使用的是开源的 EdXposed Manager，未知来源的 Manager 可能执行恶意操作
忽略
我的设备
v93.0
v0.5.2.2_4683-master_4683 (YAHFA)
v4.6.2 (46200)
Android 11 (Red Velvet Cake, API 30)
Google Google Pixel 5
AArch64 Processor rev 14 (aarch64)
arm64-v8a (arm64)
SELinux 状态: Enforcing
引导验证已启用
引导验证（dm-verity）防止设备在修改系统分区时引导

        - 表示用的是，更加新的93的EdXposed的Xposed的API版本
        - 注：更新的API版本，支持其他更新的API接口
            - 但是同时会兼容旧版本Xposed的API接口的
    - 如果是之前旧的 `Android 4.4` 的安卓手机，则往往Xposed的API版本写的是 `54` ，测试就要此处设置为
        - `xposedminversion=53`
            - 表示支持 `<= Android 5.0` ，因为支持 `Android 4.x` 的Xposed的API版本是 `54`

附录：旧版本Xposed插件的 `app/src/main/AndroidManifest.xml` 配置

另外贴出，之前，针对于旧版本 `Android 4.4` 的安卓手机

- 当时是 `Android 4.4.2` 的 `Nox夜神模拟器`



- 对应adb连接并查看设备

```
crifan@licrifandeMacBook-Pro ~ adb connect 127.0.0.1:62001
connected to 127.0.0.1:62001
crifan@licrifandeMacBook-Pro ~ adb devices
List of devices attached
127.0.0.1:62001    device
```

的 `src/main/AndroidManifest.xml` 的配置：

```
<manifest       android="http://schemas.android.com/apk/res/android"
          tools="http://schemas.android.com/tools"
    package="com.crifan.crifanxposedmodule">

    <!--android:debuggable="true"-->
    <!--tools:ignore="HardcodedDebugMode"-->
    <!--android:exported="true"-->

    <application
                allowBackup="true"
                icon="@mipmap/ic_launcher"
                label="@string/app_name"
                roundIcon="@mipmap/ic_launcher_round"
                supportsRtl="true"
                theme="@style/AppTheme"
    >
        <!-- 是否是xposed模块，xposed根据这个来判断是否是模块 -->
        <meta-data
                    name="xposedmodule"
                    value="true" />

        <!-- 模块描述，显示在xposed模块列表那里第二行 -->
        <meta-data
                    name="xposeddescription"
                    value="crifan测试XPosed模块开发" />

        <!-- 最低xposed版本号。用最新的：82
            详见：
                https://github.com/rovo89/XposedBridge/wiki/Using-the-Xposed-Framework-API
                https://github.com/rovo89/XposedBridge/wiki/Development-tutorial
                https://raw.githubusercontent.com/rovo89/XposedBridge/art/app/doclib/api/changelog.txt
            但会报错：该模块需要较新版本的XposedBridge（82），因此无法激活
            所以还是改为 53
         -->
        <meta-data
                    name="xposedminversion"
                    value="53" />

    </application>
</manifest>
```

供参考。

给 `app/build.gradle` 中加上xposed包的依赖

改动文件：`app/build.gradle`

目的：让Gradle下载相关xposed的jar包（和相关源码）

核心改动：加上：

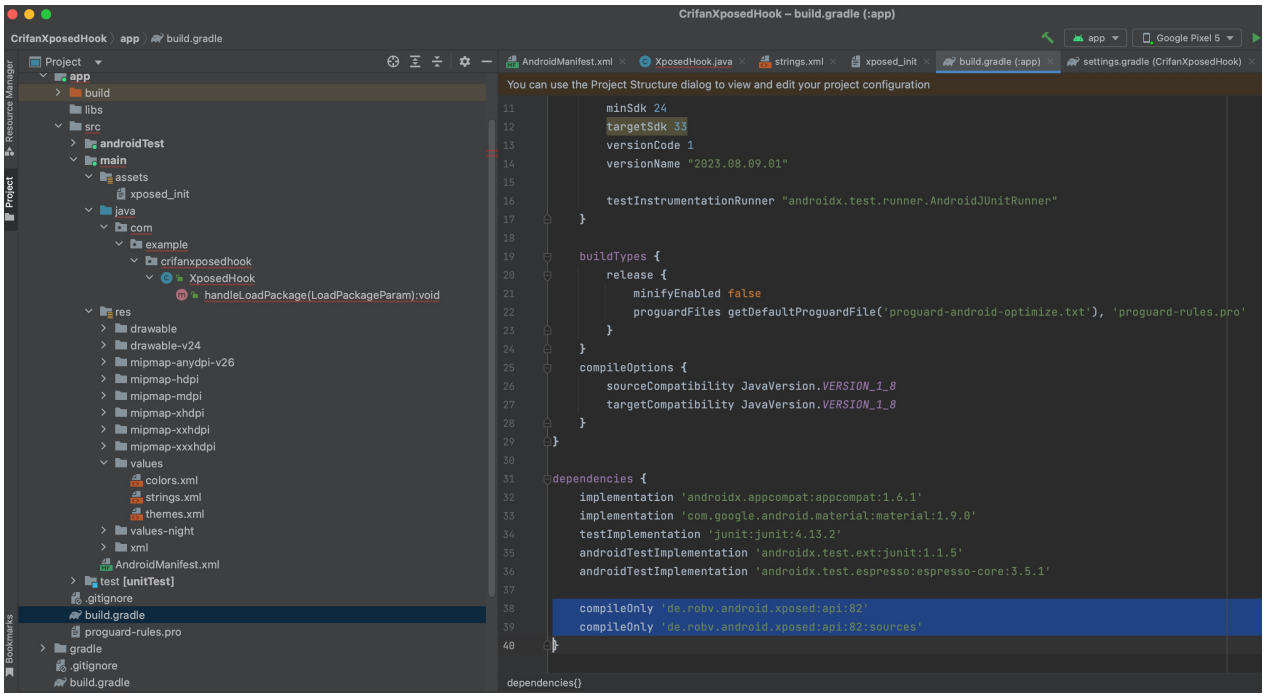
```
        versionName "2023.08.09.01"

    compileOnly 'de.robv.android.xposed:api:82'
    compileOnly 'de.robv.android.xposed:api:82:sources'
```

改为：

`app/build.gradle`

```
plugins {
```

```
    id 'com.android.application'
}

android {
    namespace 'com.example.crifanxposedhook'
    compileSdk 33

    defaultConfig {
        applicationId "com.example.crifanxposedhook"
        minSdk 24
        targetSdk 33
        versionCode 1
        versionName "2023.08.09.01"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
}

dependencies {
    implementation 'androidx.appcompat:appcompat:1.6.1'
    implementation 'com.google.android.material:material:1.9.0'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'

    compileOnly 'de.robv.android.xposed:api:82'
    compileOnly 'de.robv.android.xposed:api:82:sources'
}
```



参数说明：

- xposed相关
  - `compileOnly 'de.robv.android.xposed:api:82'`
    - xposed的jar包依赖，版本是用最新的 `82`
      - 和之前的 `AndroidManifest.xml` 中的 `xposedminversion=82` 是对应的

    - `compileOnly 'de.robv.android.xposed:api:82:sources'`
        - xposed的源码包
    - 注：
        - xposed官网推荐的 `provided` （而不要用 `compile` ）已废弃，所以改用推荐的：`compileOnly`
            - `compileOnly` ：表示代码编译会调用到，但是编译后输出的apk不会包含XPosed的库
                - 这样就不会导致报错：`java.lang.IllegalAccessError Class ref in pre-verified class resolved to unexpected implementation`
- app相关
    - `versionName "2023.08.09.01"`
        - app的版本号
            - 在app列表详情页和EdXposed的Xposed插件列表中可以看到

附录：旧版本Android Studio中app的Gradle配置

对于之前的旧版本的 `Android Studio v4.2.2` 来说：

app的Gradle配置： `app/build.gradle`

主要是加了： `compileOnly 'de.robv.android.xposed:api:53'`

内容为：

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 29
    defaultConfig {
        applicationId "com.crifan.crifanxposedmodule"
        minSdkVersion 19
        targetSdkVersion 29
        versionCode 3
        versionName "1.2"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'

    compileOnly 'de.robv.android.xposed:api:53'
}
```

全局Gradle配置文件：settings.gradle

项目级别的=全局的Gradle配置文件： `settings.gradle`

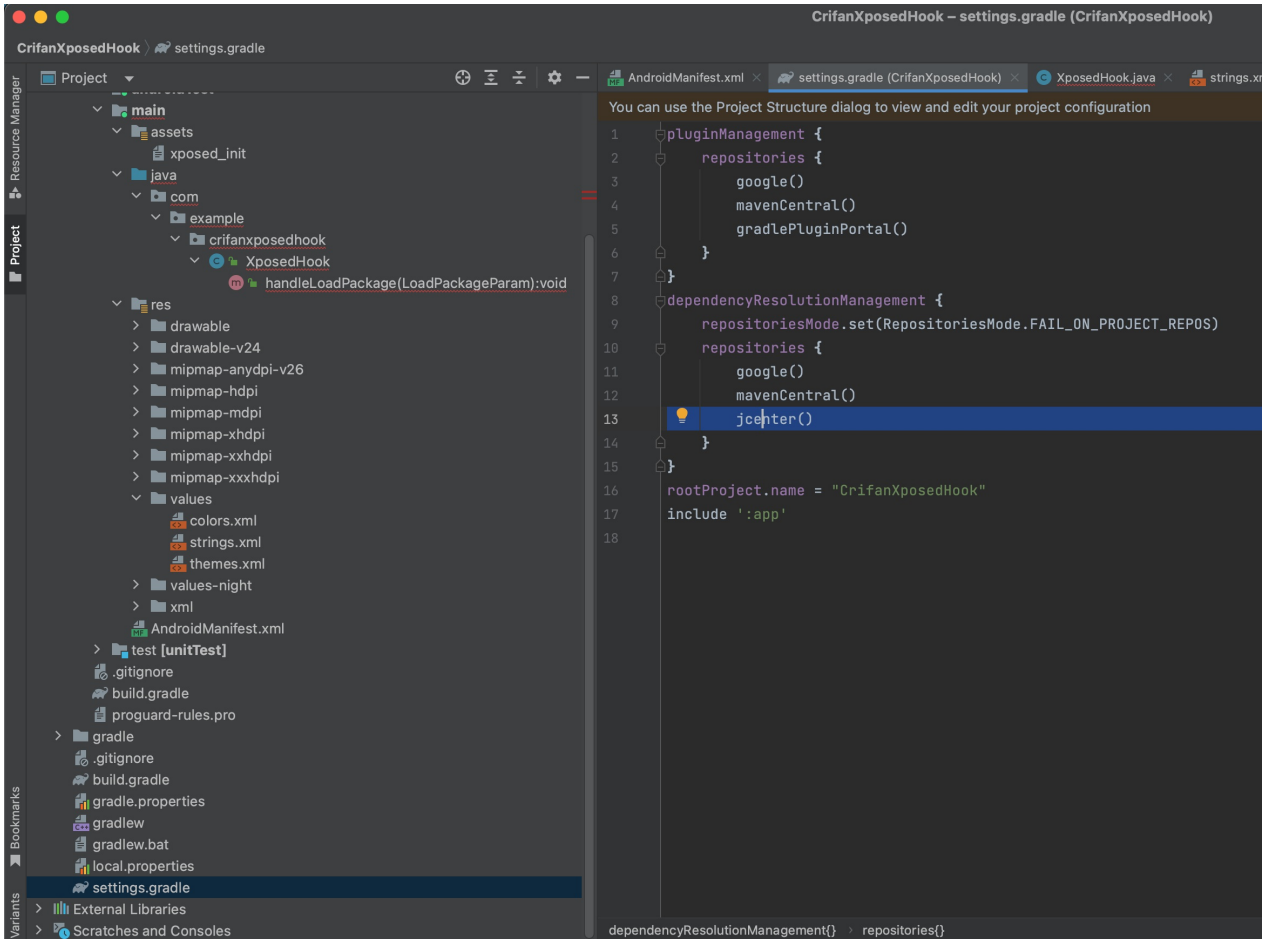
目的：加入xposed包所在的源jcenter的依赖

改动：给 `dependencyResolutionManagement` 的 `repositories` 加上 `jcenter()`

变成：

```
pluginManagement {
    repositories {
        google()
        mavenCentral()
```

```
        gradlePluginPortal()
    }
}
dependencyResolutionManagement {
    repositoriesMode.set(RepositoriesMode.FAIL_ON_PROJECT_REPOS)
    repositories {
        google()
        mavenCentral()
        jcenter()
    }
}
rootProject.name = "CrifanXposedHook"
include ' app'
```



参数说明：

- `jcenter()`
  - xposed的官网的jar包的源，就放在了 `jcenter`
    - Gradle去sync同步后，会去下载，之前加入的依赖
      - `compileOnly 'de.robv.android.xposed:api:82'`
      - `compileOnly 'de.robv.android.xposed:api:82:sources'`
    - 分别对应的jar包地址：
      - https://jcenter.bintray.com/de/robv/android/xposed/api/82/api-82-sources.jar
      - https://jcenter.bintray.com/de/robv/android/xposed/api/82/api-82.jar

附录：旧版本Android Studio中全局Gradle配置

对于之前的旧版本的 `Android Studio v4.2.2` 来说：

系统全局的： `build.gradle` 配置：

核心改动： `buildscript` 和 `allprojects` 中的 `repositories` 中，都加上了： `jcenter()`

内容为：

```
// Top-level build file where you can add configuration options common to all sub-projects/modules.

buildscript {
    repositories {
        google()
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build gradle 7.0.0'

        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        google()
        jcenter()
    }
}

task clean(type  Delete) {
    delete rootProject.buildDir
}
```

调试参数设置

Launch改为Nothing

此处，由于：

- 新建项目时，就选择了 `No Activity`
- 且此处项目中的确也没有Activity

所以需要去：

把项目调试配置中的：

`Android Studio` -> `Run` -> `Edit Configuration` -> `Android App` -> `app` -> `General` -> `Launch Options` ->从默认的：

- `Launch`：`Default Activity`

改为：

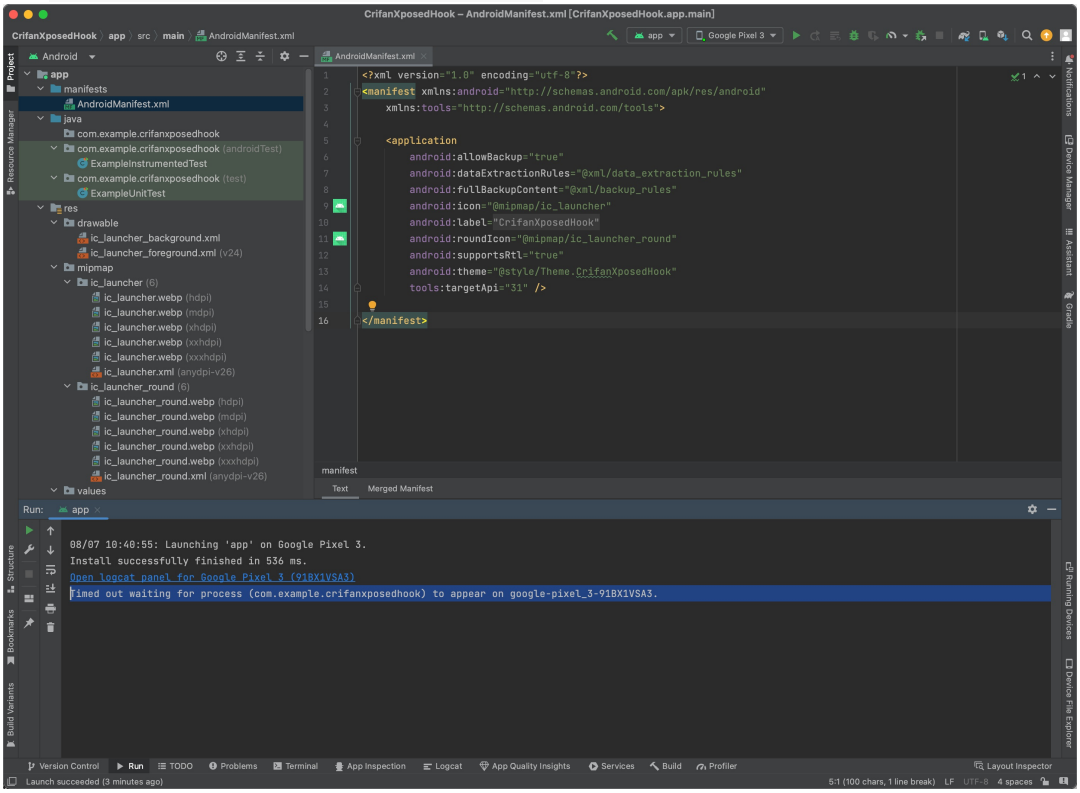
- `Launch`：`Nothing`

说明：

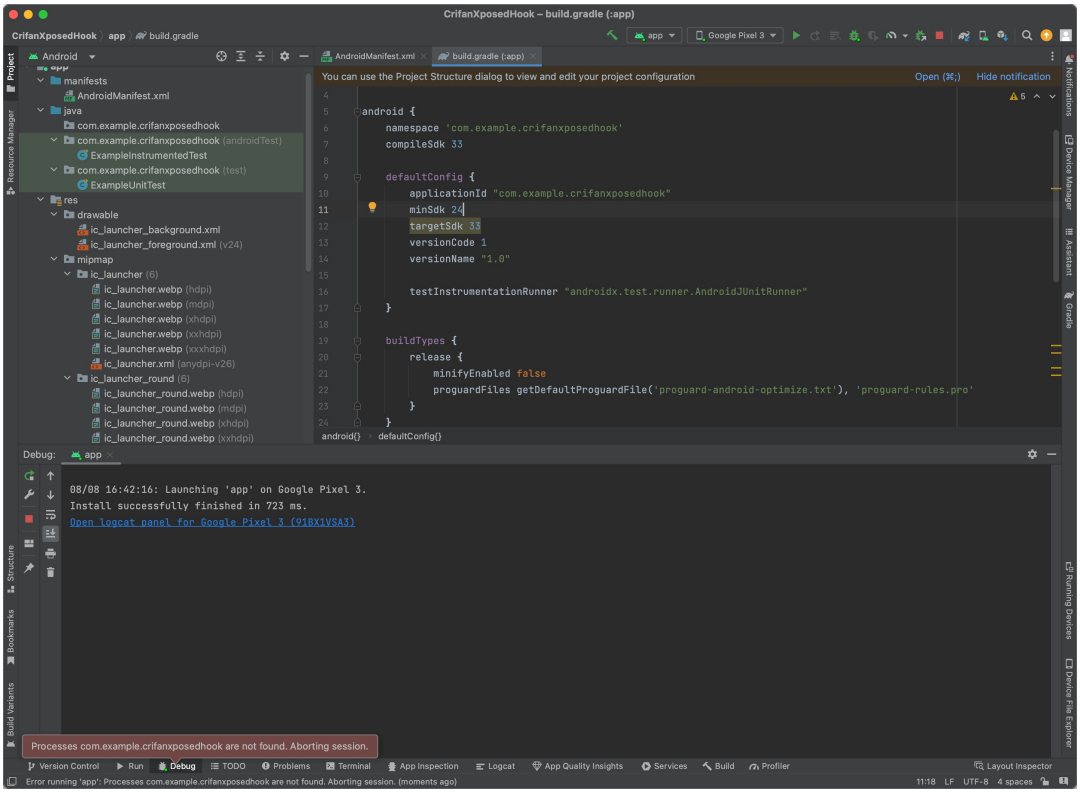
- 如果不改，则默认去Debug调试，会去尝试启动Default Activity，但是由于此处没有Activity=页面，导致报错

```
Could not identify launch activity: Default Activity not found
Error while Launching activity
Failed to launch an application on all devices
```

- 改了后，后续点击Debug调试按钮去调试时
  - 最后会报错
    - 要么是：`Timed out waiting for process xxx to appear on`



    - 要么是：`Processes xxx are not found Aborting session`



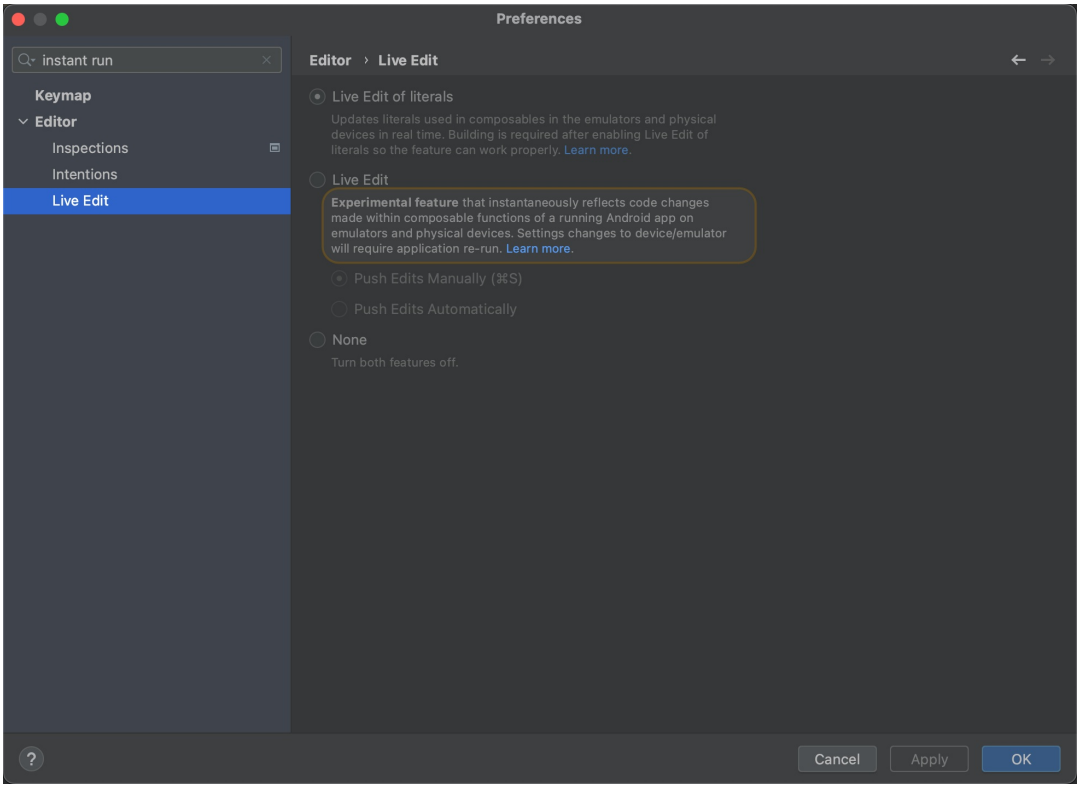
- 原因：感觉是，安卓中，Activity=页面=Process=进程
  - 而当前Xposed插件的项目中，的确没有Activity，所以没有进程可供调试，所以报错

其他

无需关闭Instant Run

此处Android Studio的版本是 `Android Studio Flamingo | 2022.2.1 Patch 2`，此处：

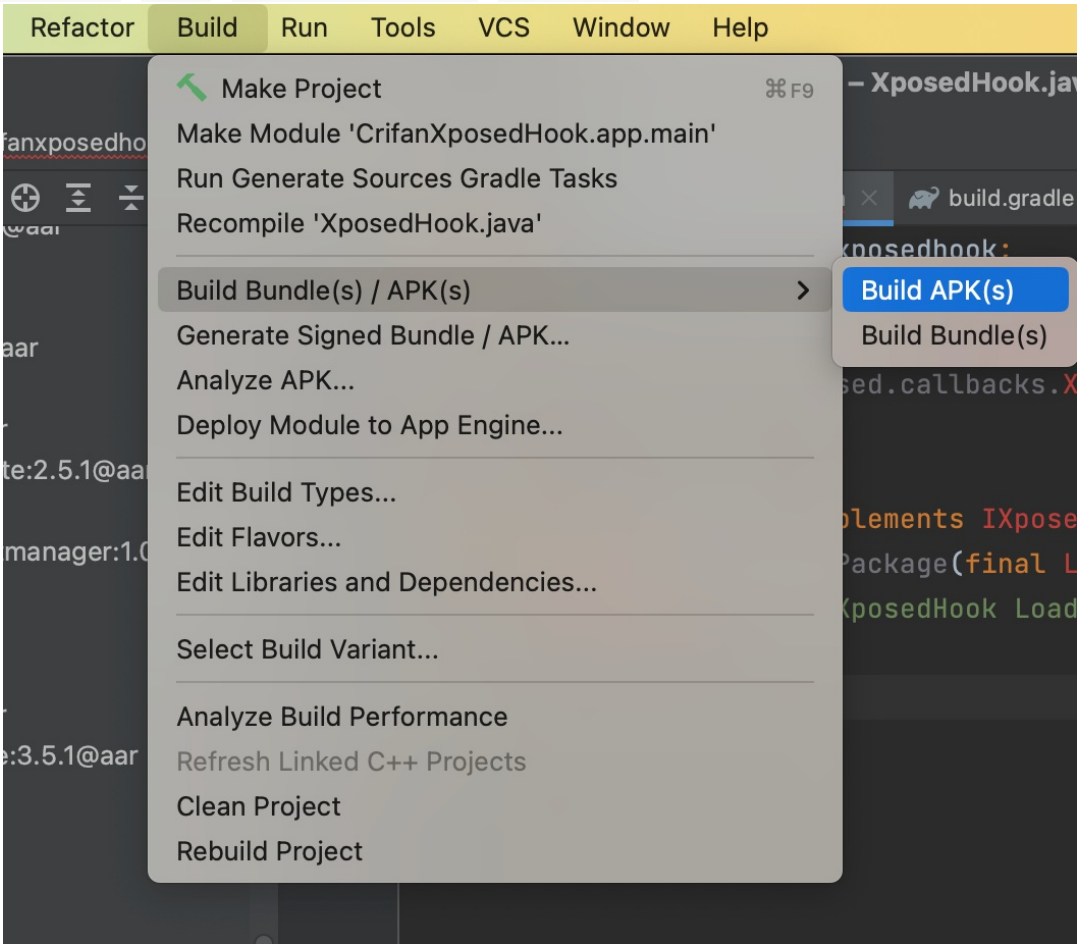
- 已经没有官网提示的Instant Run了
  - 最多算是有个，或许相关的，但是默认没有开启的
    - `Live Edit`



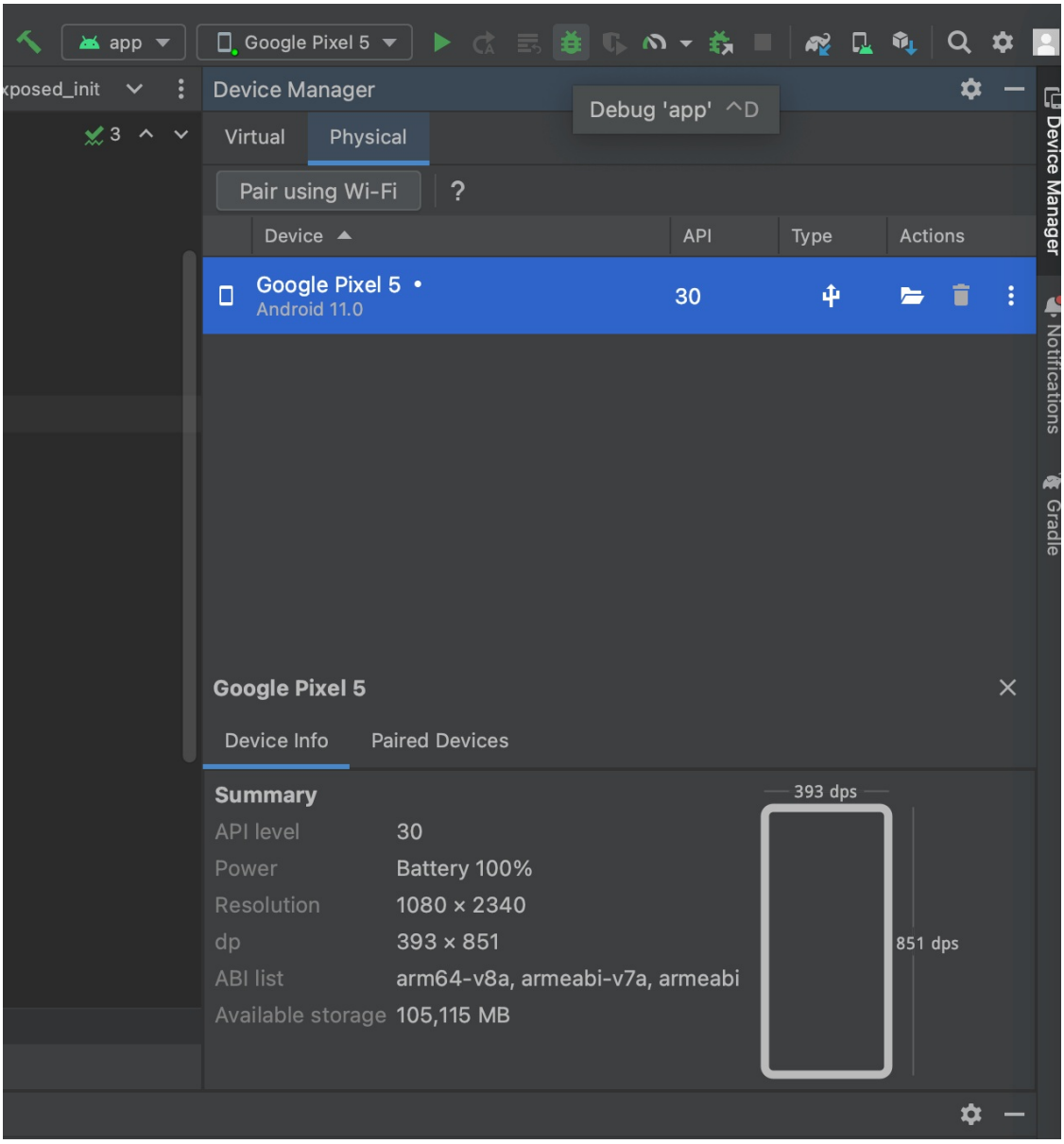
- 所以无需去：关闭 `Instant Run` 。

编译安装Xposed插件apk

接着再去

- 手动编译生成apk
  - 编译项目=编译出Xposed插件的安卓apk = Build apk
    - `Android Studio` -> `Build` -> `Build Bundle(s)/APK(s)` -> `Build APK(s)`
      - 

    - 即可编译出Xposed插件=类似于普通安卓app的apk文件
  - 再手动下载到安卓手机中，手动去安装apk
- 或者直接自动的：编译+安装apk
  - 点击 `Android Studio` 中的 `Debug app` （或运行 `Run` ）按钮



- 即可自动完成
  - 编译出apk
  - 安装apk到安卓手机
  - 尝试调试（或运行）
    - 注：此处实际上由于没有Activity，所以调试（或运行）会报错：
      - 找不到进程
        - `Processes are not found Aborting session` 或 `Timed out waiting for process to appear on`
      - 不过此处已实现我们的目标：安装最新版Xposed插件的apk到安卓手机了
        - 剩下的，则忽略这些错误即可

确认插件已安装和能生效

此时，`Xposed插件 = 普通的安卓apk`，已安装到安卓手机中：

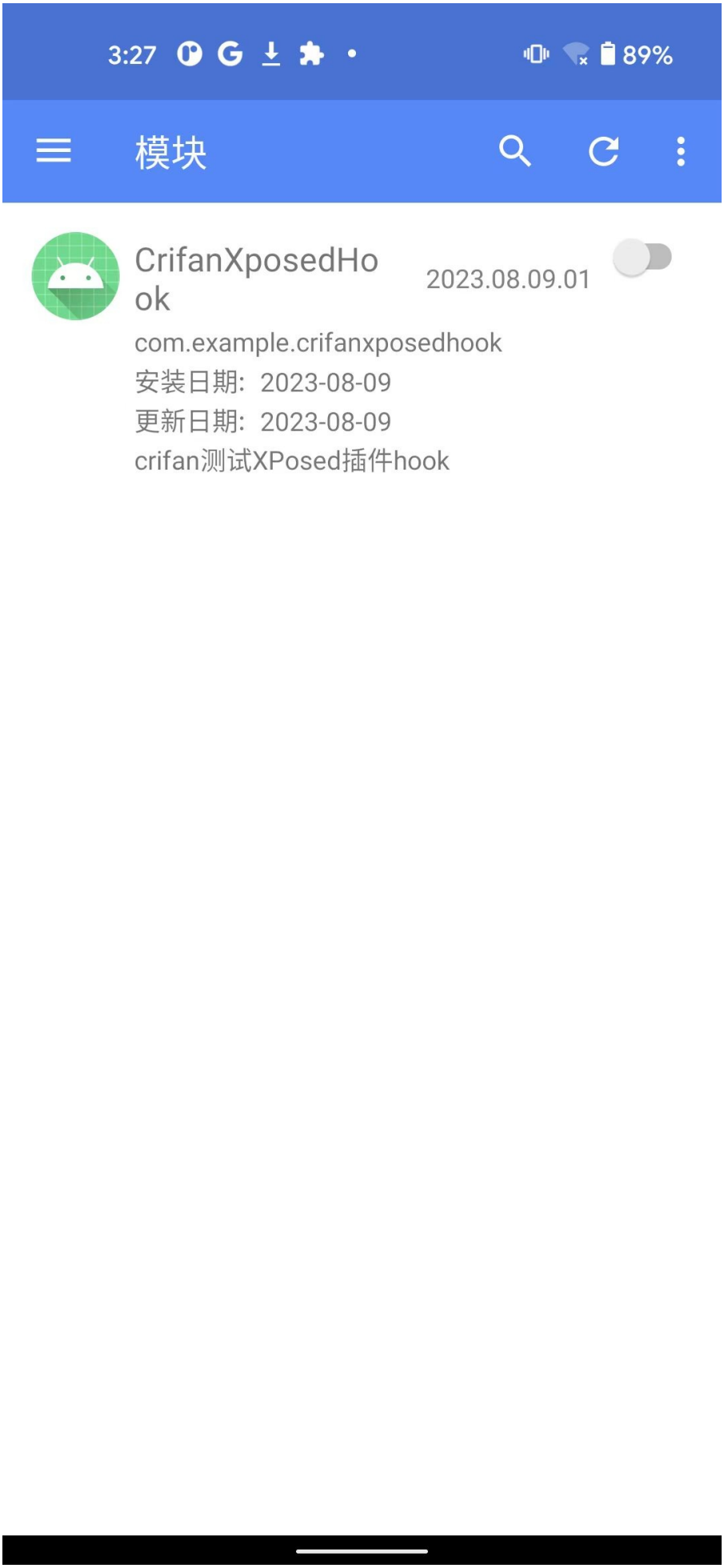
如何确认Xposed插件已安装成功

可以通过，去下面这些地方：

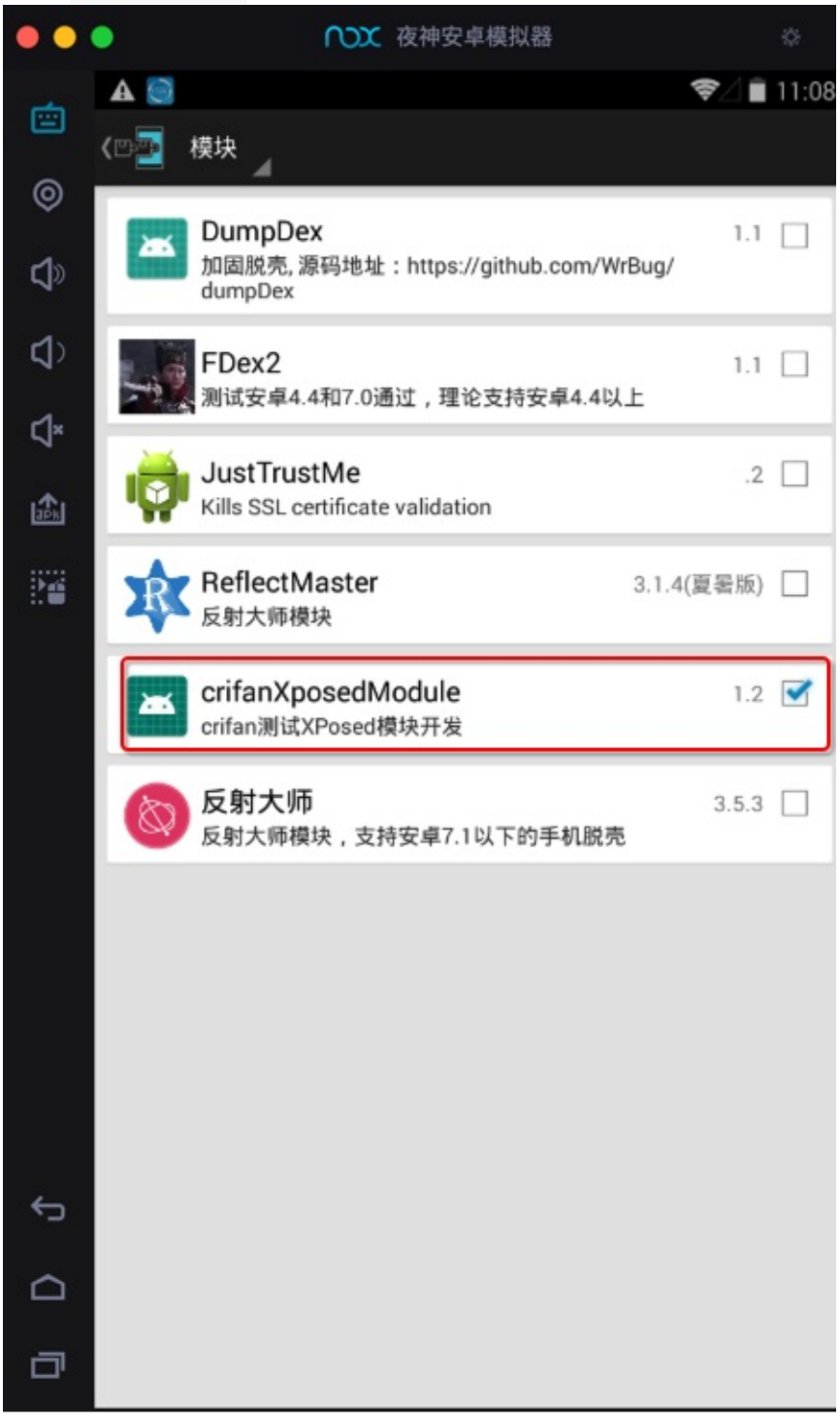
- 安卓手机中设置中系统的应用列表中
  -


3:26
89%
应用信息
CrifanXposedHook
卸载
强行停止
通知
每周大约 0 条通知
权限
未请求任何权限
存储和缓存
已使用 6.36 MB 内部存储空间
移动数据和 WLAN
未使用任何数据流量
设备使用时间
今天使用了0分钟
电池
自上次充满电后已使用 0%
默认打开
没有默认操作
版本 2023.08.09.01

- Xposed框架中模块列表中
  - 此处EdXposed
    -


3:27
89%
模块
CrifanXposedHook
2023.08.09.01
com.example.crifanxposedhook
安装日期: 2023-08-09
更新日期: 2023-08-09
crifan测试XPosed插件hook

- 对比：旧的 `Xposed Installer` 中的Xposed模块列表中，显示效果是这种

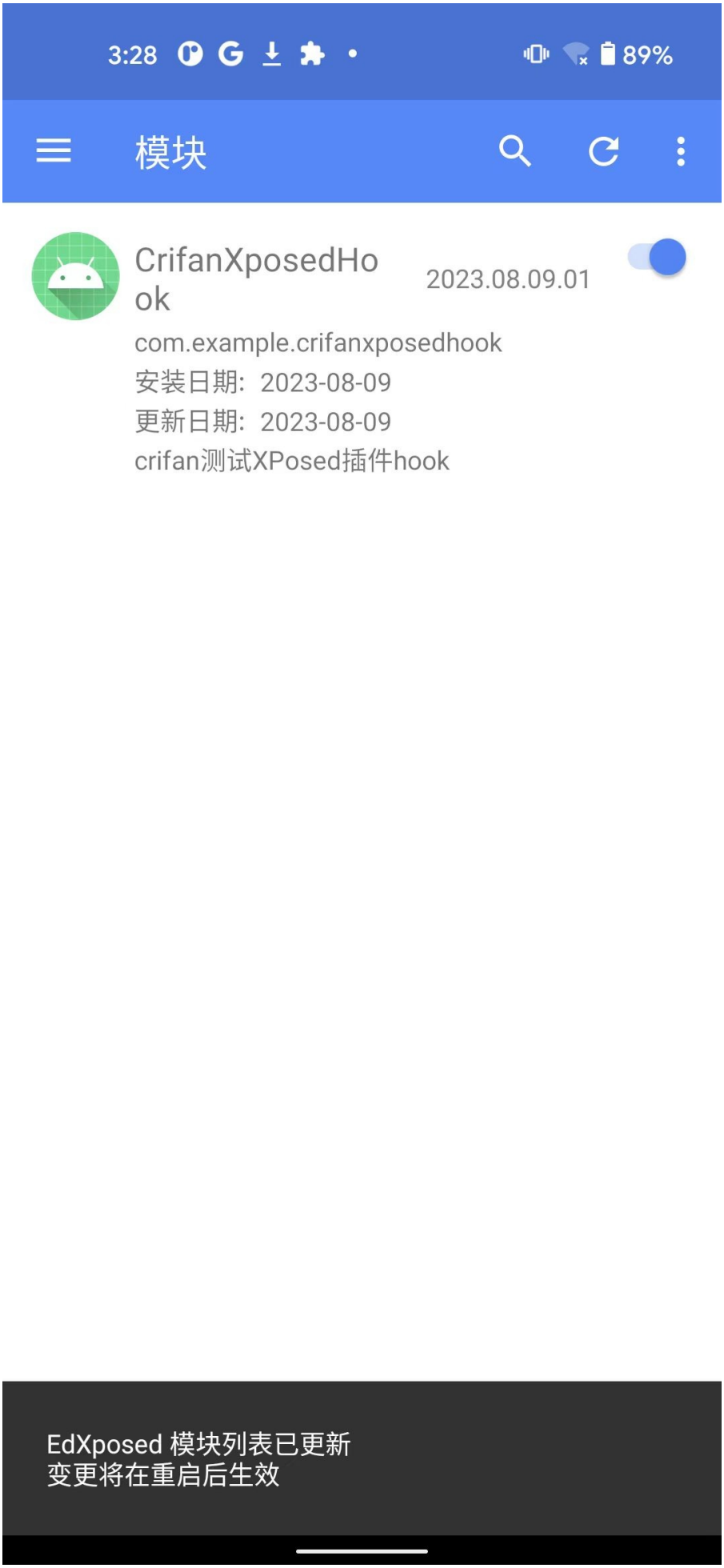


能看到：自己的Xposed插件=普通安卓app，则说明：Xposed插件已成功安装

如何确认Xposed插件正常工作已生效

先开启插件

此处EdXposed中看到的自己的Xposed插件，右边有个开关，点击开启：


3:28
89%
模块
CrifanXposedHook
2023.08.09.01
com.example.crifanxposedhook
安装日期: 2023-08-09
更新日期: 2023-08-09
crifan测试XPosed插件hook
EdXposed 模块列表已更新
变更将在重启后生效

注：开启插件时，底部会出现提示：`EdXposed模块列表已更新，变更将在重启后生效`

（根据提示，继续）再去重启EdXposed

- 对比：旧的 `Xposed Installer` 中，新安装了Xposed插件后，需要重启Android手机才能使得插件生效

即可成功开启插件。

再确认插件是否正常工作

如何确认=测试Xposed插件已生效=真正正常工作了？

先去安卓手机Pixel5中操作：随便打开任意一些普通的安卓app即可

然后去找找，能看到输出对应hook的log日志，就说明插件正常生效了

问题变成：如何查看Xposed插件的hook的log日志

目前已知主要有2种方式：

- Xposed框架中的日志
- Android Studio中的LogCat

Xposed框架中的日志

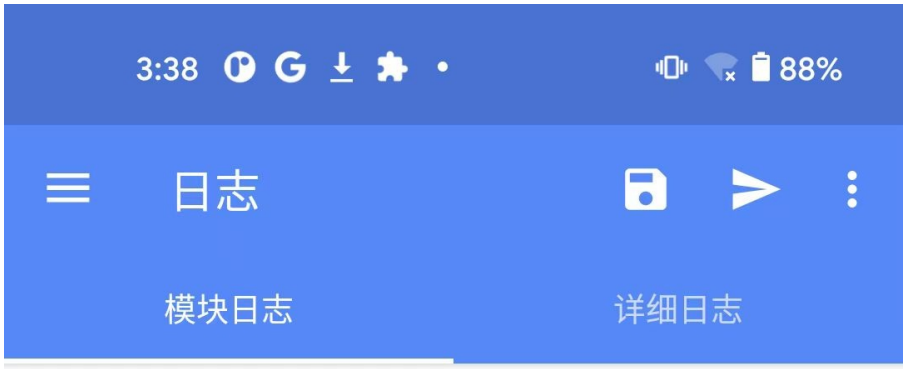
对于此处的：EdXposed

`EdXposed` -> `日志` -> `模块日志` ：


3:45
87%
EdXposed
Xposed Hide
模块
下载
日志
设置
关于
详细日志

就可以看到hook的log日志：


3:37
88%
日志
模块日志
详细日志
EdXposed Log
Powered by Log Catcher
QQ support group: 855219808
Telegram support group: @Code_Of_MeowCat
Telegram channel: @EdXposed
--------- beginning of information
Manufacturer: Google
Brand: google
Device: redfin
Product: redfin
Model: Pixel 5
Fingerprint: google/redfin/redfin:11/RQ3A.211001.001/7
ROM description: redfin-user 11 RQ3A.211001.001 764197
Architecture: arm64-v8a
Android build: RQ3A.211001.001
Android version: 11
Android sdk: 30
EdXposed version: v0.5.2.2_4683-master
EdXposed api: 93
Riru version: v25.3.4.r399.84f7084 (399)
Riru api: 25
Magisk: 25.2:MAGISK (25200)
--------- beginning of main
--------- beginning of system
--------- beginning of crash
08-09 15:31:13.253 28102 28102 I EdXposed-Bridge: Load
08-09 15:31:13.692 28102 28102 I EdXposed-Bridge: Lo
08-09 15:31:13.913 28102 28102 I EdXposed-Bridge: Xpos
08-09 15:31:21.951 28342 28342 I EdXposed-Bridge: Load
08-09 15:31:22.379 28342 28342 I EdXposed-Bridge: Lo
08-09 15:31:22.571 28342 28342 I EdXposed-Bridge: Xpos
08-09 15:32:05.516 28102 28158 I EdXposed-Bridge: Xpos
08-09 15:37:02.714 1472 1472 I EdXposed-Bridge: Load
08-09 15:37:02.876 1472 1472 I EdXposed-Bridge: Lo
08-09 15:37:02.911 1472 1472 I EdXposed-Bri pos




```
.001/7641976:user/release-keys
7641976 release-keys
```

```
Loading modules from /data/app/~~lakNq25Fyxv5X7_V5isv
  Loading class com.example.crifanxposedhook.XposedHo
XposedHook Loaded app: com.google.android.apps.safety
Loading modules from /data/app/~~lakNq25Fyxv5X7_V5isv
  Loading class com.example.crifanxposedhook.XposedHo
XposedHook Loaded app: com.tencent.mm
XposedHook Loaded app: com.google.android.gms
Loading modules from /data/app/~~lakNq25Fyxv5X7_V5isv
  Loading class com.example.crifanxposedhook.XposedHo
XposedHook Loaded app: org.meowcat.edxposed
```

其中的：

- XposedHook Loaded app: com.google.android.apps.safetyhub
- XposedHook Loaded app: com.tencent.mm
- XposedHook Loaded app: com.google.android.gms
- XposedHook Loaded app: org.meowcat.edxposed.manager

就表示，我们之前的Xposed的hook代码

```
XposedBridge.log("XposedHook Loaded app: " + lpparam.packageName);
```

真正工作了：能hook到对应的app了，输出对应的log了。

- 导出全部日志

另外，点击右上角纸飞机的按钮，可以导出当前全部的 `.txt` 日志文件：

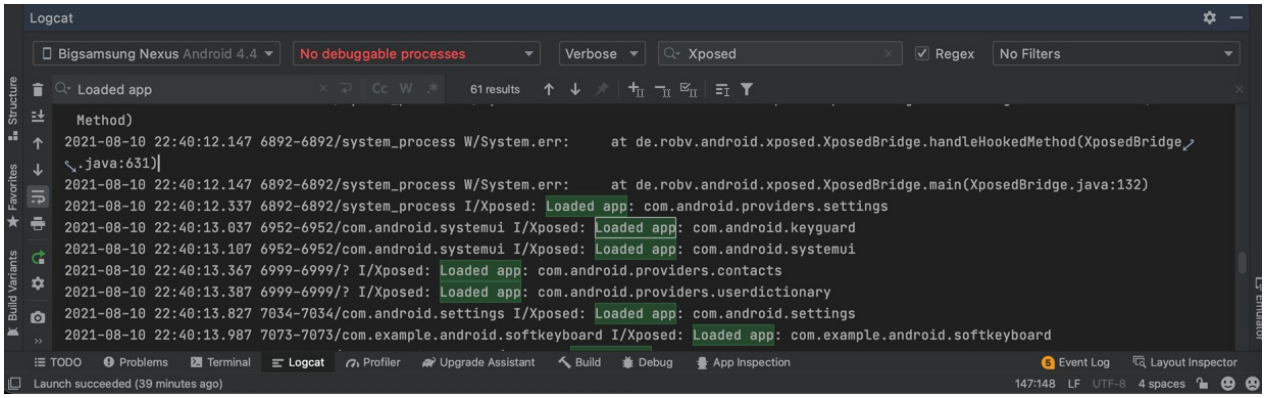
- `EdXposed_Modules_20230809_153813.txt`

```
EdXposed Log
Powered by Log Catcher
QQ support group: 855219808
Telegram support group: @Code_Of_MeowCat
Telegram channel: @EdXposed
--------- beginning of information
Manufacturer: Google
Brand: google
Device: redfin
Product: redfin
Model: Pixel 5
Fingerprint: google/redfin/redfin:11/RQ3A.211001.001/7641976:user/release-keys
ROM description: redfin-user 11 RQ3A.211001.001 7641976 release-keys
Architecture: arm64-v8a
Android build: RQ3A.211001.001
Android version: 11
Android sdk: 30
EdXposed version: v0.5.2.2_4683-master
EdXposed api: 93
Riru version: v25.3.4.r399.84f7084 (399)
Riru api: 25
Magisk: 25.2:MAGISK (25200)
--------- beginning of main
--------- beginning of system
--------- beginning of crash
08-09 15:31:13.253 28102 28102 I EdXposed-Bridge: Loading modules from /data/app/~~lakNq25Fyxv5X7_V5iswfQ==/com.
example.crifanxposedhook-nubXoNVT2XbwkrSmpZB8tQ==/base.apk
08-09 15:31:13.692 28102 28102 I EdXposed-Bridge:   Loading class com.example.crifanxposedhook.XposedHook
08-09 15:31:13.913 28102 28102 I EdXposed-Bridge: XposedHook Loaded app: com.google.android.apps.safetyhub
08-09 15:31:21.951 28342 28342 I EdXposed-Bridge: Loading modules from /data/app/~~lakNq25Fyxv5X7_V5iswfQ==/com.
example.crifanxposedhook-nubXoNVT2XbwkrSmpZB8tQ==/base.apk
08-09 15:31:22.379 28342 28342 I EdXposed-Bridge:   Loading class com.example.crifanxposedhook.XposedHook
08-09 15:31:22.571 28342 28342 I EdXposed-Bridge: XposedHook Loaded app: com.tencent.mm
08-09 15:32:05.516 28102 28158 I EdXposed-Bridge: XposedHook Loaded app: com.google.android.gms
08-09 15:37:02.714  1472  1472 I EdXposed-Bridge: Loading modules from /data/app/~~lakNq25Fyxv5X7_V5iswfQ==/com.
example.crifanxposedhook-nubXoNVT2XbwkrSmpZB8tQ==/base.apk
08-09 15:37:02.876  1472  1472 I EdXposed-Bridge:   Loading class com.example.crifanxposedhook.XposedHook
08-09 15:37:02.911  1472  1472 I EdXposed-Bridge: XposedHook Loaded app: org.meowcat.edxposed.manager
```

对比：旧版Xposed：`XPosed Installer` 中，看到的Xposed插件的hook的log日志是：

Android Studio中的LogCat

（之前旧版）`AS` -> `LogCat` -> `filter`：`XPosed`，可以查看到hook日志：



Xposed插件demo项目

此处，去把上述整个的Xposed插件的demo的项目，上传到Github仓库：

- crifan/CrifanXposedHook: Xposed Module plugin demo project of Android Studio
  - https://github.com/crifan/CrifanXposedHook.git

供参考。

Xposed心得

TODO：

- 心得
  - 【已解决】Xposed如何hook混淆后的安卓应用中的类名和函数名

安卓破解可能会涉及到Xposed

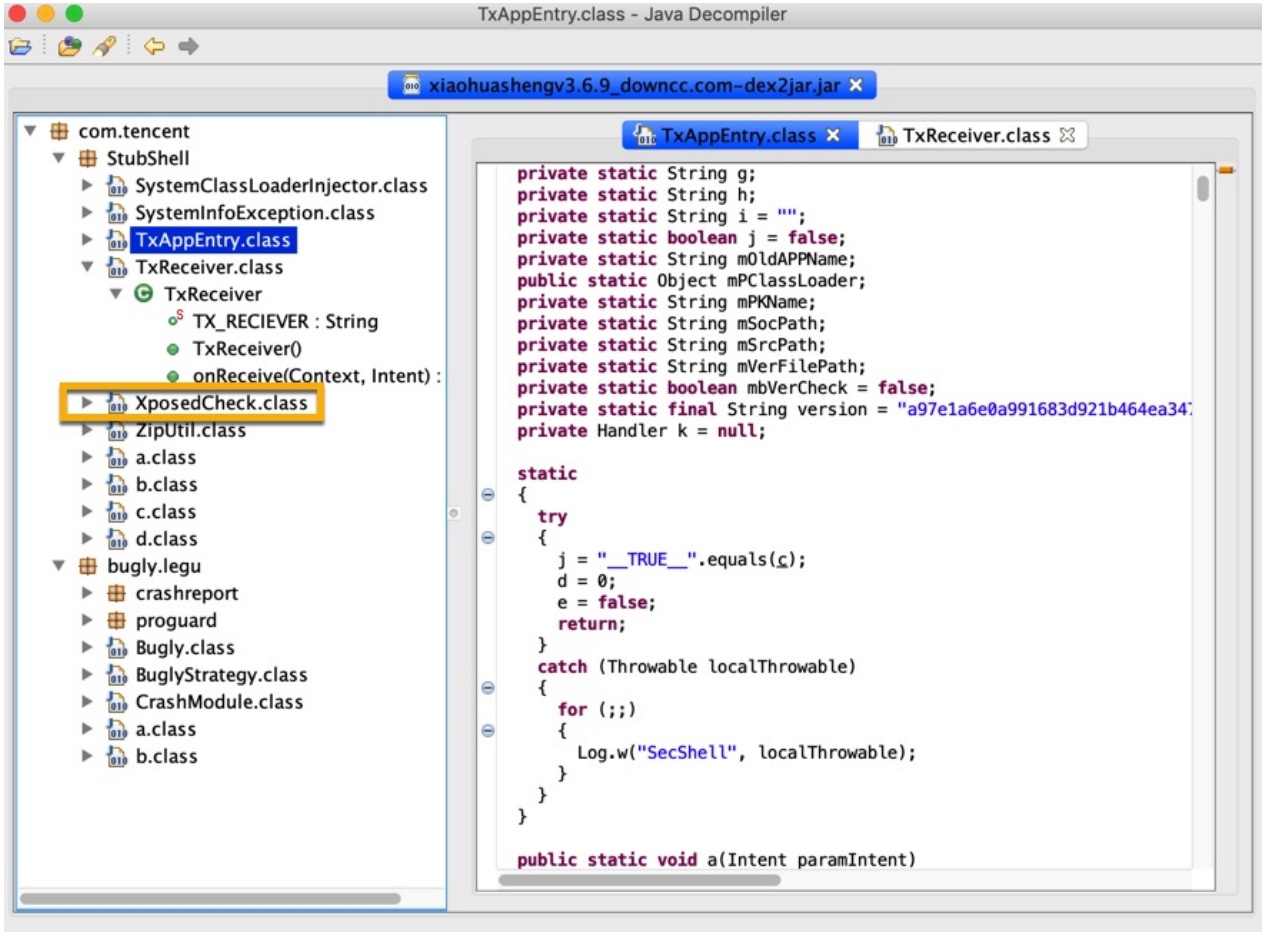
注意到：

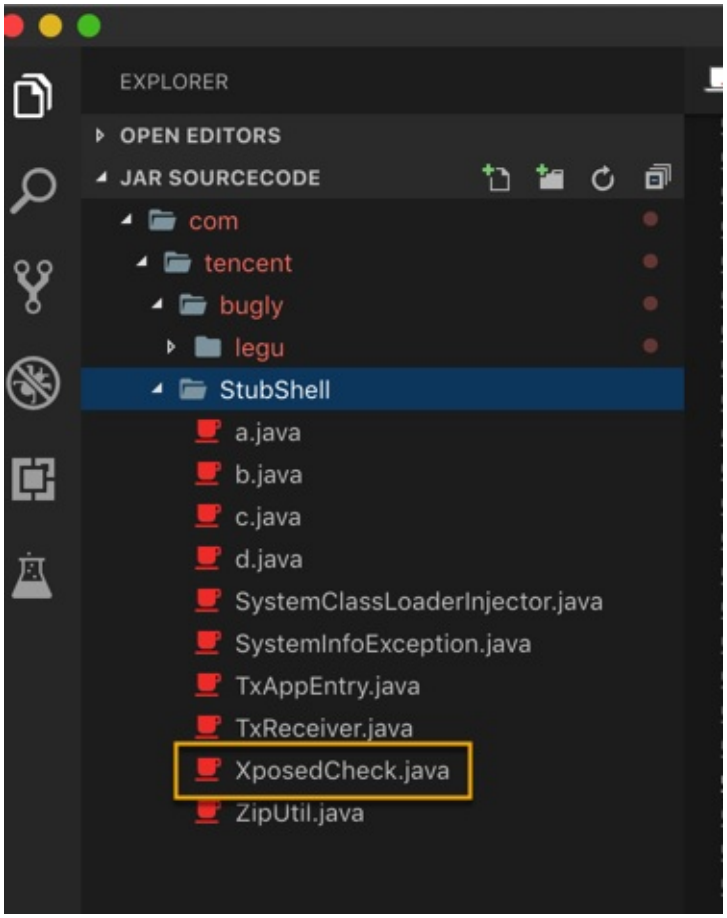
【已解决】mac版JD-GUI查看并导出jar包的java源代码

和：

【已解决】搞懂安卓app混淆和加固常见做法和相关逻辑

中都有个：`XPosedCheck`





-》看来是安卓apk加固方案内部用到了：去check检测XPosed相关的逻辑

-》说明软件加固的破解中，有些是借助XPosed去破解安卓的

-》实际情况的确也是，XPosed之类的框架，是帮助破解安卓的利器

附录

下面列出相关参考资料。

参考资料

- 【未解决】编写EdXposed插件去hook调试Android应用YouTube
- 【未解决】给安卓手机Google Pixel3去安装Xposed
- 【已解决】参考XPosed官网教程创建最基本的XPosed插件的安卓apk
- 【已解决】AS调试安卓Xposed插件app
- 【已解决】安卓逆向编写Xposed插件：创建安卓app项目
- 【已解决】AS中给Xposed插件初始化配置和写hook代码
- 【已解决】AS中Xposed插件代码报错：Cannot resolve symbol de
- 【无需解决】AS中Xposed插件代码报错：Cannot resolve symbol IXposedHookLoadPackage
- 【已解决】对于EdXposed的Pixel3如何设置Xposed插件的xposedminversion
- 【已解决】Android Studio的gradle出现警告：WARNING Configuration provided is obsolete and has been replaced with compileOnly
- 【无需解决】AS调试安卓app报错：Processes are not found Aborting session
- 【无需解决】AS中调试安卓app报错：Timed out waiting for process to appear on
- 【已解决】安卓手机Pixel5中EdXposed中确认Xposed插件是否安装成功和是否生效
- 【已解决】搞懂XPosed插件开发流程并开发出一个最基本的XPosed插件
- 【已解决】Android Studio调试没有Activity的XPosed插件时报错：Error running app Default Activity not found
- 【已解决】Xposed插件加载报错：dalvikvm had used a different IXposedHookLoadPackage during pre-verification
- 【已解决】XPosed插件开发期间xposedminversion应该改成什么值
- 【已解决】Android 11中Magisk中安装EdXposed插件
- 【已解决】在Android 11中安装EdXposed框架
- 【已解决】Android 11中Magisk中安装Riru插件
- 【已解决】Android 11中EdXposed Manager的下载安装和基本使用
- 【已解决】给root后的Android 11的Google Pixel中安装XPosed框架
- 【记录】google pixel3中查看已安装过的Magisk和EdXposed
- 【已解决】Android 11的Magisk中安装EdXposed 0.5.2.2结果失败：请先从Magisk Manager中安装Riru Installation failed
- 【已解决】Android13的Pixel5安装LSPosed
- 【已解决】Magisk中安装LSPosed后去打开LSPosed的app
- 【记录】LSPosed的app的界面和功能
- 【记录】给Magisk安装MagiskHide替代方案：Shamiko模块
- 【记录】LSPosed的详细日志内容
- 【已解决】Magisk中的：Zygisk
- 【记录】Xposed框架继承者：EdXposed、LSPosed、Tai Chi
- 【未解决】XposedInstaller中安装Xposed插件：[MM-O] Xdebuggable
- 【记录】下载安装试用XPosed模块：GravityBox
- 【已解决】XPosed模块：GravityBox
-
- 【模块框架】LSPatch - 免Root模块框架小白向安装使用指南 - 知乎 (zhihu.com)
- Android Lspatch(模块工具) 0.5.1(385) - 果核剥壳 (ghxi.com)
- 安卓应用的安全和破解
- 好用的安卓模拟器：夜神Nox
- Releases · ElderDrivers/EdXposedManager (github.com)
- Releases · RikkaApps/Riru (github.com)
- Releases · ElderDrivers/EdXposed (github.com)
- [BUG] Error detect riru · Issue #873 · ElderDrivers/EdXposed (github.com)
- [BUG] Error detect riru · Issue #873 · ElderDrivers/EdXposed (github.com)
- [原创]Xposed 安装记录-『Android安全』-看雪安全论坛
- Android APK脱壳--腾讯乐固、360加固一键脱壳 | 辉天神龙
- Difference between Xposed and VirtualXposed · android-hacker/VirtualXposed Wiki

- How does VirtualXposed work · android-hacker/VirtualXposed Wiki
- 不需要 Root，也能用上强大的 Xposed 框架：VirtualXposed - 少数派
- Xposed框架概况 - Xposed框架中文站
- Xposed框架 - 维基百科，自由的百科全书
- Xposed框架中文站 - 超多Xposed框架模块介绍与下载
- 【Android】Xposed 框架解析 - 简书
- IXposedHookLoadPackage | Xposed Framework API
- Pixel 5 Root, 安装Xposed框架, 解锁5G教程_android_Shadestart-开源项目管理 (csdn.net)
- Zygisk: What is it? How to Download and Use it?
- 使用 GravityBox 自定义您的 Android 界面以及更多功能 | 安卓帮助 (androidayuda.com)
- 什么是GravityBox，Android 上最好的Xposed 模块？ 安卓帮助 (androidayuda.com)
- 重力工具箱(GravityBox)如何使用手机软件软件教程_脚本之家 (jb51.net)