目录

iOS逆向：iOS15越狱

- 最新版本：`v1.2`
- 更新时间：`20230819`

简介

介绍iOS越狱中的iOS 15+之后的越狱工具和事项。包括palera1n、XinaA15、Dopamine等。先是iOS15+的越狱概览，然后是北背景知识，包括rootless和rootful、iPhone机型信息、免签安装ipa文件，其中包括TrollStore和，Sideloadly；以及详细介绍iPhone中安装TrollStore；接着介绍palera1n，包括越狱前的前提条件、注意事项和说明、工具的版本、文档和资料，和越狱中的palera1n的rootful越狱概述和详解，以及常见问题、help语法、loader app和越狱后的ssh、如何恢复越狱、常见问题；接着介绍XinaA15，包括越狱前的历史版本和下载地址，和越狱中的安装XinaA15、用XinaA15越狱，以及越狱后的界面和功能；接着介绍Dopamine；最后给出参考资料。

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/ios_re_ios15_jailbreak: iOS逆向：iOS15越狱

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

- iOS逆向：iOS15越狱 book.crifan.org
- iOS逆向：iOS15越狱 crifan.github.io

离线下载阅读

- iOS逆向：iOS15越狱 PDF
- iOS逆向：iOS15越狱 ePub
- iOS逆向：iOS15越狱 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

iOS15越狱概览

- iOS 15+的越狱
  - 说明：截至20230628，iOS 15+的越狱工具大多还不算很稳定，都处于开发中(in development)
    - 所以主要是给开发者(Developer)用，不建议普通iPhone用户使用
  - 概述
    - 基本可用的
      - palera1n
        - 支持 `A8-A11` 的 `iOS 15.0` - `iOS 16.5` 的 `rootful` 和 `rootless` 越狱
      - XinaA15
        - 支持 `A12+` 的 `iOS 15.0` - `iOS 15.1.1` 的 `rootless` 越狱
      - Dopamine
        - 支持 `A12-A15, M1` 的 `iOS 15.0` - `iOS 15.4.1` 的 `rootless` 越狱
    - 其他可能有用的
      - `Cheyote`
      - `Fugu15`
      - `Blizzard Jailbreak`

iOS15+越狱背景知识

对于 `iOS 15+` 的越狱，有些额外的背景知识需要了解，下面介绍一下。

rootful普通越狱 vs rootless无根越狱

- 概述
  - `rootful` 越狱=普通越狱=**有根**越狱
    - rootfs可写，包括根目录/也可以写
      - palera1n中也叫： `fakefs-rootful`
  - `rootless` 越狱 = **无根**越狱
    - rootfs只读，只有/var可写
- 详解
  - 之前的 `普通越狱` = `成熟越狱` = `有根越狱` vs `无根越狱` = `rootless jailbreak`
    - 之前的 `普通越狱` = `成熟越狱` = `有根越狱` ： `iOS 15` 之前
      - 越狱工具（有机会利用系统漏洞实现）：对于iOS的根文件系统 `rootfs` ，去读写
        - 重新挂载根文件系统（为读写）
    - `无根越狱` = `rootless jailbreak` ： `iOS 15` 之后
      - iOS
        - 内部机制增加了：（Apple的） `SSV` = `Signed System Volume`
          - 细节：对于原本的，被iOS设计为只读readonly的各种系统目录
            - 如果尝试去写入，则会由于filter过滤 和 nullifying而被拒绝 -》即，不允许写入
        - 效果：不允许对于rootfs去写 == `sealed ROOT File System`
      - 越狱工具
        - 无法改动/写入iOS的根文件系统 `rootfs`
        - 想要实现越狱，则只能：避免写入=改动iOS 15的 `rootfs`
        - 所以：新的iOS 15之后的越狱工具，都是 `rootless jailbreak` = `无根越狱`
          - rootless=对于rootfs没有写入的权限=没法改动rootfs根文件系统
          - 特殊
            - 不过有2个特殊的目录，可以写入：
              - `/var`
              - `/private/preboot`
            - 而新的iOS 15之后的越狱系统，基本上都是利用这个机制，去实现越狱的效果
              - 即：把越狱相关工具和内容，都放到 `/var` （和 `/private/preboot` ）中
        - 说明
          - rootless jailbreak，并不表示没有root用户=root user
            - 你还是可以以 `root user` 身份去操作：SSH连接到设备，修改（ `/var` 和 `/private/preboot` ）中的文件的
          - 由此使得，取消越狱=卸载越狱=恢复原始设备，就变得很容易
            - 因为只是涉及到 `/var` 和 `/private/preboot` ，不涉及到整个rootfs的改动和恢复，所以很快很方便
          - 相对来说：绕过越狱检测，相对容易一些
            - 估计指的是，只是针对 `/var` ，而不是整个系统，所以简单点?
      - rootless有哪些影响
        - **部分插件** `tweak` 失效
          - 之前的各种(依赖于旧的rootfs存放文件的)插件tweak：就失效了
            - 需要tweak插件作者去更新，才能支持新的 `rootless jailbreak`
        - `bootstrap` 也失效了
          - bootstrap：用于启动阶段，安装各种Unix工具，包管理器等等
          - 现在需要用改用：兼容rootless的bootstrap = rootless (compatible) bootstrap
            - 比如
              - `Procursus`
                - 【整理】iOS越狱相关：Procursus
              - `Elucubratus`
        - 文件管理器filesystem browsers： `Filza` , `SSH` , and `Apple File Conduit`

- 基本可用：只是无法write修改系统目录中文件了（可以读取read和执行execute）
  - 可以写入 `/var` 和 `/private/preboot` 目录

iPhone机型信息

越狱之前，需要搞清楚，常见的iPhone设备的信息，尤其是arm的架构，尤其是最新的 `A12+` 芯片所支持的 `ARMv8.3` 的 `arm64e` 。

因为后续比如 `palera1n` 等越狱时，会涉及到。要先搞清楚：

- ARM Architecture: `arm64e`
  - iPhone 13 / iPhone 13 Pro
    - SoC: `A15`
  - iPhone 12 / iPhone 12 Pro
    - SoC: `A14`
  - iPhone 11 / iPhone 11 Pro / iPhone SE (2nd generation)
    - SoC: `A13`
  - iPhone XS / iPhone XR
    - SoC: `A12`
- ARM Architecture: `arm64`
  - iPhone X / iPhone 8
    - SoC: `A11`
  - iPhone 7
    - SoC: `A10`
  - iPhone 6s / iPhone SE (1st generation)
    - SoC: `A9`
  - iPhone 6
    - SoC: `A8`
  - iPhone 5s
    - SoC: `A7`
- ARM Architecture: `armv7s`
  - iPhone 5c / iPhone 5
    - SoC: `A6`
- ARM Architecture: `armv7`
  - iPhone 4S
    - SoC: `A5`
  - iPhone 4
    - SoC: `A4`
- ARM Architecture: `armv6`
  - iPhone 3GS
    - SoC: `APL0298`
  - iPhone 2G
    - SoC: `APL0098`

搞懂iPhone机型信息有何用?

这样才能在iOS逆向期间，对于涉及到ARM架构的时候，有所了解底层的含义：

比如：

iOS 13的AppleIDAuthSupport库所支持的iPhone机型

不同机型所对应的ARM的架构是：

- armv6

- armv7
- armv7s
- arm64
- arm64e

而iOS的很多的Framework库，本身支持足够多的iOS的机型，所以会看到：

iOS的Framework库的 `tbd` 信息中，包括了支持的多个的 `ARM` 的 `arch` 架构

比如：

- iOS 13的AppleIDAuthSupport.tbd
  - iPhoneOS13.0.sdk/System/Library/PrivateFrameworks/AppleIDAuthSupport.framework/AppleIDAuthSupport.tbd

```
....
archs:                  [ armv7, armv7s, arm64, arm64e ]
platform:               ios
...
install-name:           '/System/Library/PrivateFrameworks/AppleIDAuthSupport.framework/AppleIDAuthSupport'
```

    - iOS 13.0的AppleIDAuthSupport.framework库，支持多个ARM的arch架构：armv7, armv7s, arm64, arm64e
      - 意思就是，这个库，支持如下机型的iPhone
        - armv7 的 `iPhone 4/4S`
        - armv7s 的 `iPhone 5/5c`
        - arm64 的 `iPhone 5s/6/6s/SE(1st generation)/7/8/X`
        - arm64e 的 `iPhone XS / XR / 11/11 Pro/SE (2nd generation)/12/12 Pro/13/ 13 Pro`

arm64和arm64e的区别

才能搞懂，arm64和arm64e的区别：

XinaA15中的libsubstrate.dylib的不同版本

- `XinaA15` 越狱后的 `A12` 的 `iPhone11`
  - 有2个 `libsubstrate.dylib`
    - `/private/preboot/3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B762F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus/xina/libsubstrate.dylib`
      - 大小：`51KB`
      - 只支持一种架构：`arm64e`
    - `/private/preboot/3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B762F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus/usr/lib/libsubstrate.dylib`
      - 大小：`218KB`
      - 支持2种架构：`arm64` 和 `arm64e`

Xcode编译插件时如何支持A12芯片的iPhone11

才能明白，Xcode编译插件，去调试时：

- 报错

```
默认     14:50:37.206463+0800    Preferences     正在修复 path /private/preboot/3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B762F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus/Library/MobileSubstrate/DynamicLibraries/jailAppleAccount.dylib
默认     14:50:37.206568+0800    jailbreakd      /private/preboot/3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B762F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus/Library/MobileSubstrate/DynamicLibraries/jailAppleAccount.dylib
默认     14:50:37.207165+0800    Preferences     tweakinject 注入失败原因:dlopen(/var/Liy/Library/MobileSubstrate/DynamicLibraries/jailAppleAccount.dylib, 0x0009): tried: '/var/Liy/Library/MobileSubstrate/DynamicLibraries/jailAppleAccount.dylib' (mach-o file, but is an incompatible architecture (have 'arm64', need 'arm64e')), '/usr/local/lib/jailAppleAccount.dylib' (no such file), '/usr/lib/jailAppleAccount.dylib' (no such file), '/private/preboot/3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B762F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus/Library/MobileSubstrate/DynamicLibraries/jailAppleAccount.dylib' (mach-o file, but is an incompatible architecture (have 'arm64', need 'arm64e')), '/usr/local/lib/jailAppleAccount.dylib' (no such file), '/usr/lib/jailAppleAccount.dylib' (no such file), '/private/p
```

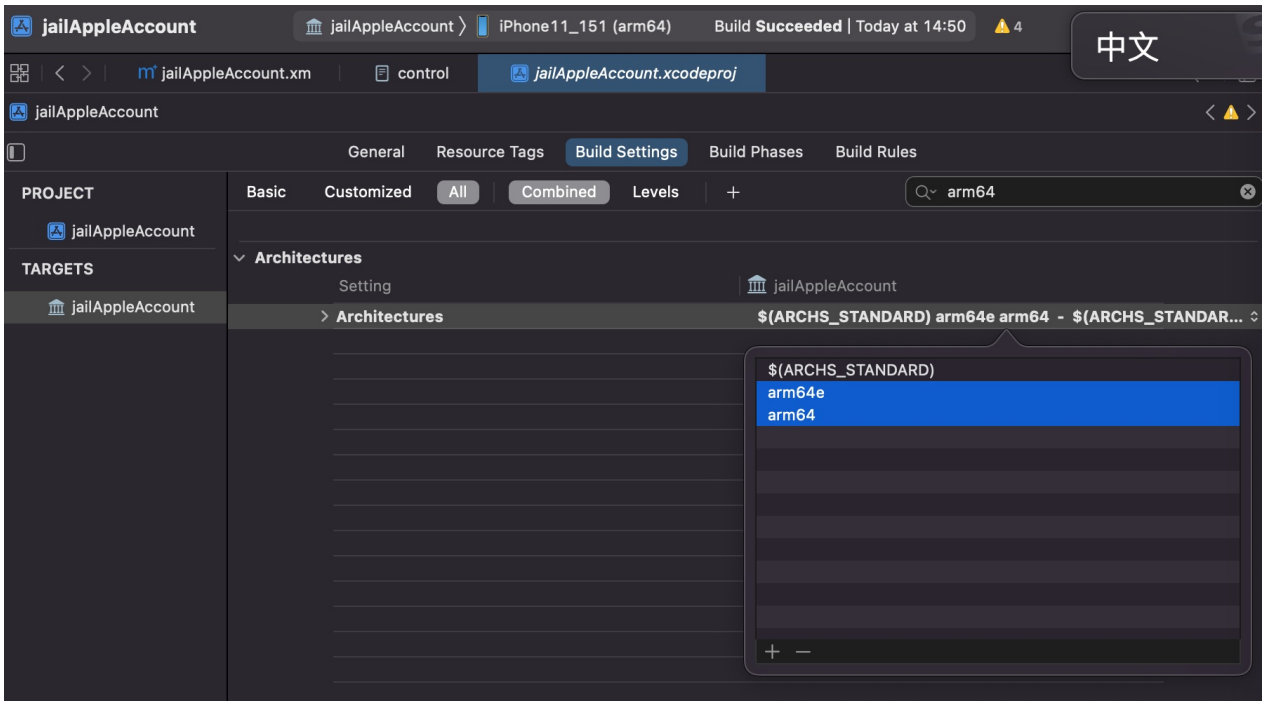
```
reboot/3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B762F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus/Library/MobileSubstrate/DynamicLibraries/jailAppleAccount.dylib' (mach-o file, but is an incompatible architecture (have
```

- 原因

是当前Xcode中的ARM的架构

- `Xcode` -> `TARGETS` -> `YourProjectName` -> `Build Settings` -> `Architectures` -> `Architectures`
  - 默认值是：`arm64`，`armv7`
    - 所以才：不支持 `arm64e` 的 `A12` 芯片的 `iPhone11`
- 解决办法

而想要让其支持 `arm64e` 的 `A12` 芯片的 `iPhone11`，则去改为：`arm64 arm64e`，即可



免签安装ipa文件

有些越狱工具，比如 `XinaA15`，推荐安装方式是： `TrollStore`，其属于：免签安装ipa的工具。

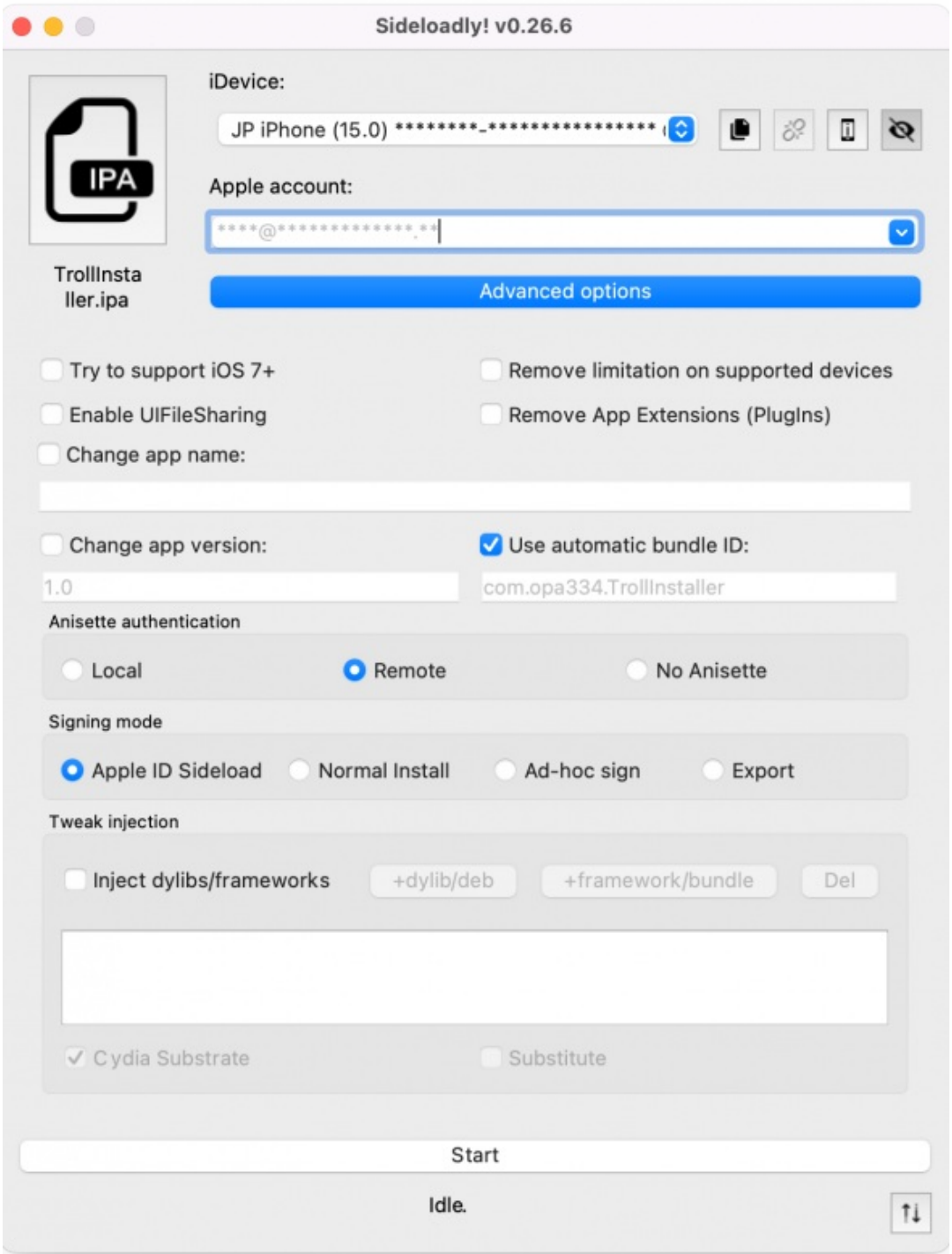
- 免签安装ipa文件
  - 背景：iOS的app的安装，需要官方有效的签名才可以。
    - 但是个人版开发者账号，默认只有7天有效期
      - 过期后，需要重新签名，很是麻烦
  - 所以：出现了很多，相关的免签名，或者是辅助的sideloe的方式，去安装ipa的工具
  - 常见免签安装ipa的工具
    - `TrollStore`
    - `Sideloadly`

TrollStore

- `TrollStore`
  - 中文名： `巨魔` = `巨魔商店`
  - 作用：永久签名工具
    - 让你可以直接安装各种ipa文件
      - 它可以在不越狱的条件随意安装IPA，且不依赖证书就能做到“永久签名”
    - 内部逻辑：绕过iOS系统的限制
      - 普通ipa，签名无法通过校验，无法安装
        - 或者是用自己的AppleID签名，但是过了默认的7天限制，需要重新签名-》很麻烦
    - 效果：使用TrollStore我们可以随便签名各种修改版的IPA、应用多开等等
      - 只要系统满足安装要求，那就不用再依赖证书，安装的IPA“永久有效”
  - 作者： `opa334`
  - logo图标
    - 

  - 官网
    - 官方的github
      - opa334/TrollStore: Jailed iOS app that can install IPAs permanently with arbitary entitlements and root helpers because it trolls Apple (github.com)
        - https://github.com/opa334/TrollStore
    - 疑似的官网
      - TrollStore - Permanently Sideload Any IPAs For Free
        - https://trollstore.app
  - 安装
    - 概述：多种安装方式
      - 通过iPhone中的Safari浏览器安装
      - 通过ipa安装
      - 等等
    - 文档
      - https://github.com/opa334/TrollStore 中的： `Installation Guides`
        - 比如适用于此处 `iOS 15.1` 的 `iPhone 11` 的
          - TrollStore/install_trollhelperota_ios15.md at main · opa334/TrollStore · GitHub

iPhone中安装TrollStore

- 有多种安装方式
  - 通过 浏览器 安装TrollStore
    - iPhone中，用Safari打开链接：https://api.jailbreaks.app/troll
    - 具体步骤详见
      - TrollStore/install_trollhelperota_ios15.md at main · opa334/TrollStore · GitHub
  - 用 `Sideloadly` 安装TrollStore
    - 对应安装包：`TrollStore Installer IPA` = `TrollInstaller.ipa`
    - 

此处最后的选择是：

- 没用：Sideloadly去安装TrollStore的ipa

    - 因为TrollStore的ipa是旧版本，而另外缺找不到最新版本的TrollStore的ipa
        - 估计是：安装了旧版本TrollStore后，也可以通过OTA升级到最新版，但是懒得去弄
- 改用：参考官网github文档，去用Safari浏览器去安装TrollStore

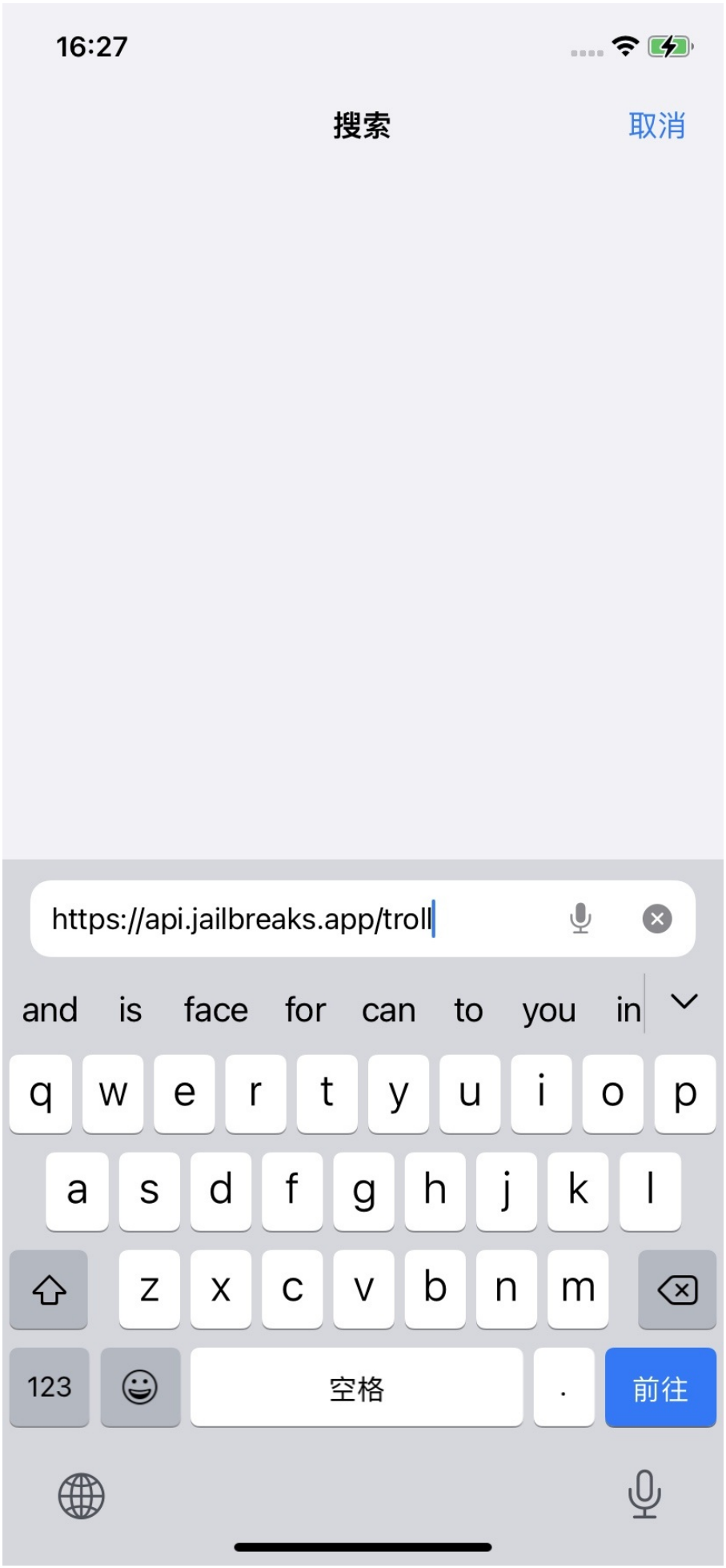
iPhone11中用Safari浏览器去安装TrollStore

核心步骤

iPhone->Safari浏览器-》打开地址 https://api.jailbreaks.app/troll -》“在iTunes中打开此页” 的弹框中：打开 -》“jailbreak.app想要安装TrollHelper”的弹框中：安装 -》桌面出现app图标JB，显示：正在安装 ->桌面上新增app：GTA Car Tracker -》点击进入GTA Car Tracker-》app标题是TrollStore Helper -》点击Install TrollStore-》稍等一会，iPhone重启-》桌面上出现：TrollStore

详细解释

- iPhone->Safari浏览器-》打开地址 https://api.jailbreaks.app/troll
    -


16:27
搜索
取消
https://api.jailbreaks.app/troll
and is face for can to you in
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
123
空格
.
前往

- -》“在iTunes中打开此页”？弹框中：打开
  -


16:27
在“iTunes”中打开此页?
取消
打开
搜索或输入网站名称

- -》“jailbreak.app想要安装TrollHelper”的弹框中：安装
  -


16:27
网页版
Bai du 百度
输入搜索词
百度一下
YY直播
百变主播 颜艺俱佳
超多精彩 尽在YY
立即观看
“jailbreaks.app”想要安装
“TrollHelper”
取消
安装
置顶
专题片
置顶
央视财经
在“球王”贝利灵柩前与他人自拍，国际足联主席因凡蒂诺…
红星新闻
2022年五个反腐细节
鲁网
大小
baidu.com

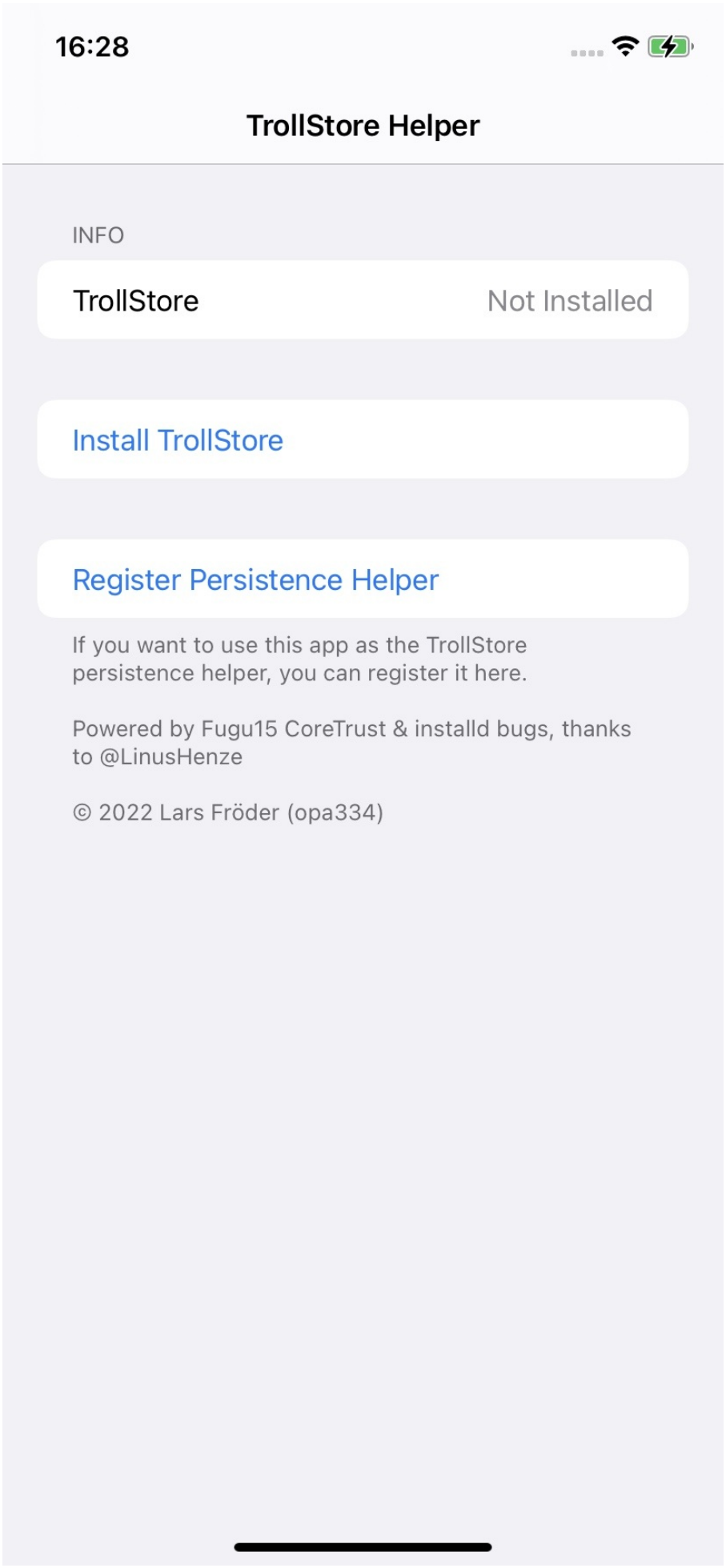
- -》桌面出现app图标JB，显示：正在安装
  -


16:27
天气
查找
快捷指令
家庭
通讯录
文件
股市
翻译
图书
iTunes Store
Watch
提示
实用工具
爱思极速版
计算器
Apple Store
电话
正在安装...
1

- ->桌面上新增app：GTA Car Tracker
  -


16:27
天气
查找
快捷指令
家庭
通讯录
文件
股市
翻译
图书
iTunes Store
Watch
提示
实用工具
爱思极速版
计算器
Apple Store
电话
GTA Car Tracker

- -》点击进入GTA Car Tracker-》app标题是TrollStore Helper
  -


16:28
TrollStore Helper
INFO
TrollStore
Not Installed
Install TrollStore
Register Persistence Helper
If you want to use this app as the TrollStore persistence helper, you can register it here.
Powered by Fugu15 CoreTrust & installd bugs, thanks to @LinusHenze
© 2022 Lars Fröder (opa334)

- -》点击Install TrollStore-》稍等一会，iPhone重启-》桌面上出现：TrollStore
  -


16:28
天气
查找
快捷指令
家庭
通讯录
文件
股市
翻译
图书
iTunes Store
Watch
提示
实用工具
爱思极速版
计算器
Apple Store
电话
GTA Car Tracker
TrollStore
1

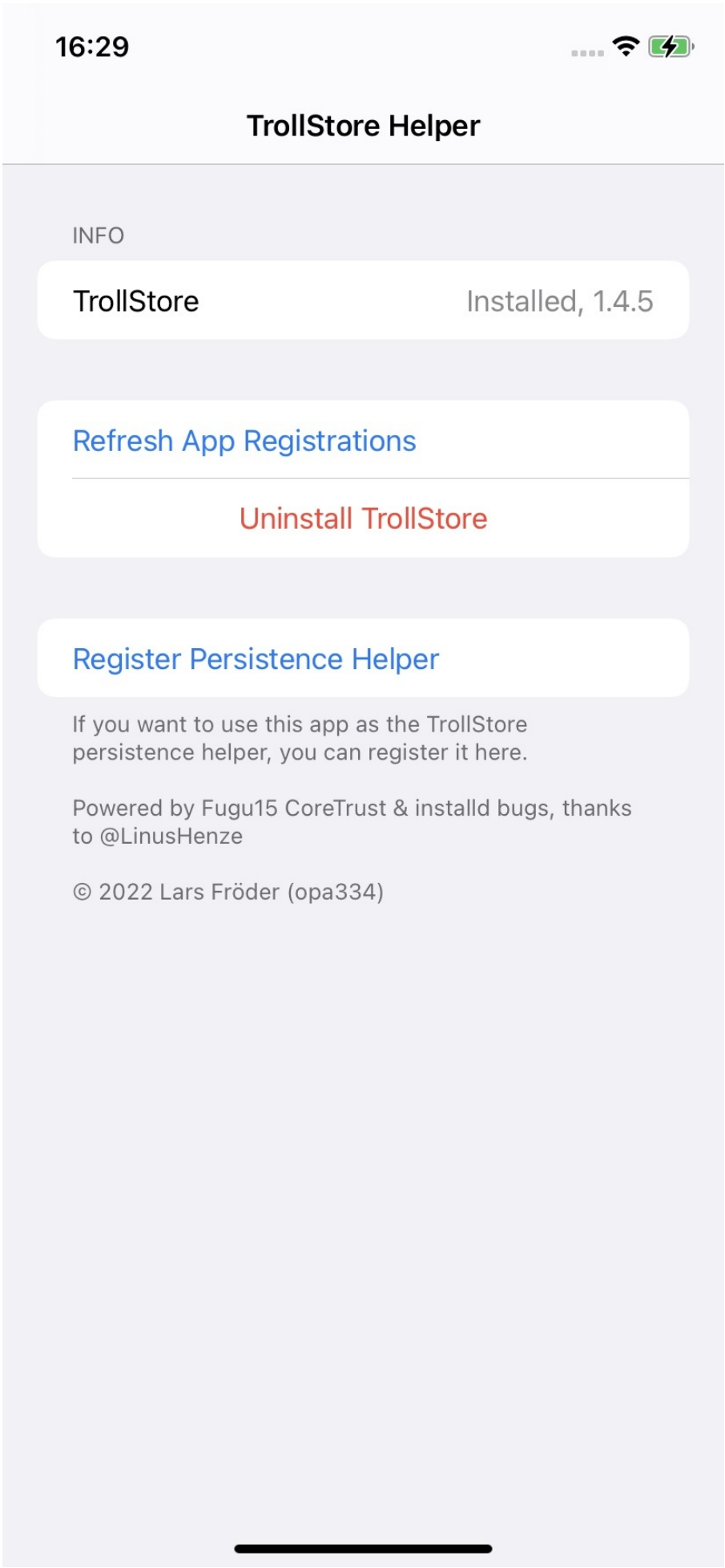
安装TrollStore后

把TrollStore设置为持续存在

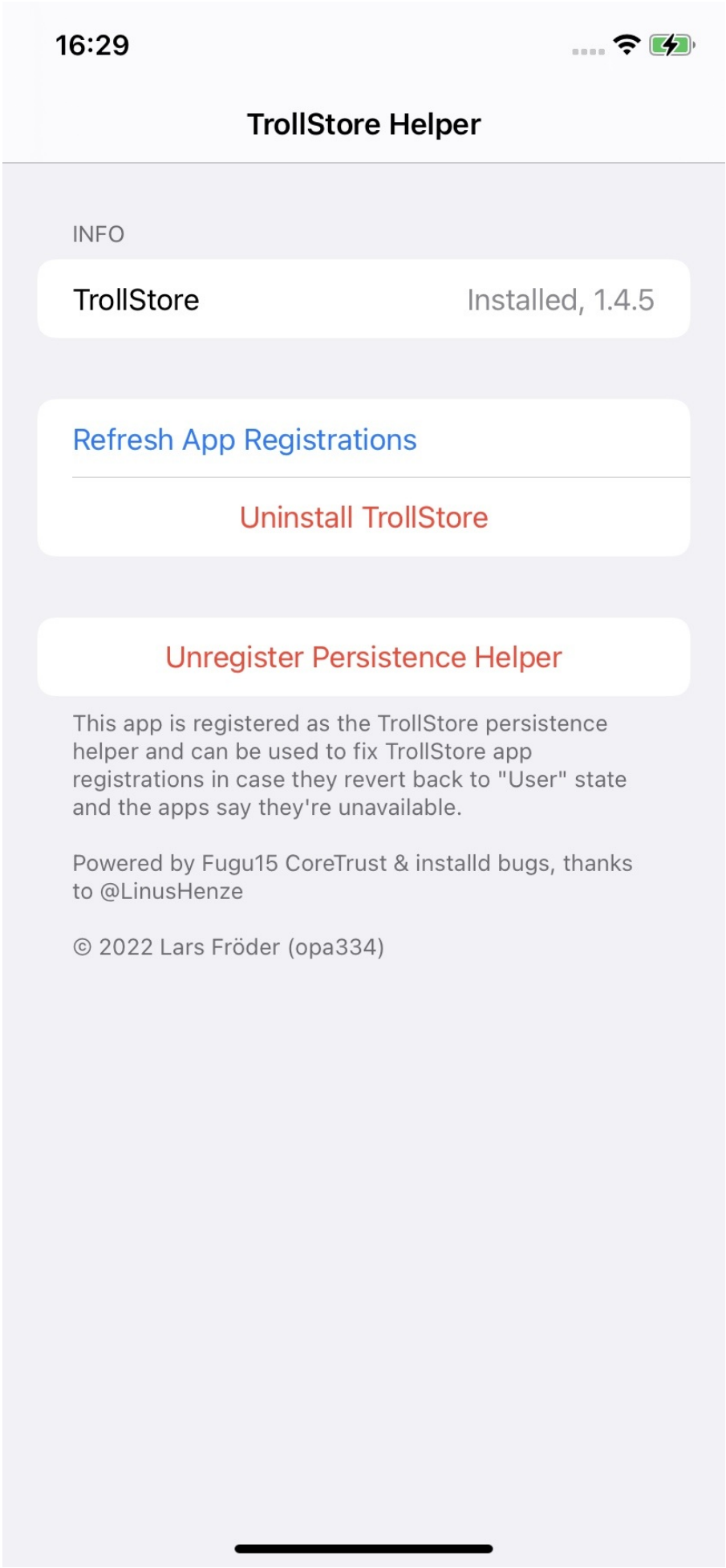
在iPhone中安装了TrollStore后，为了使后续系统图标刷新等操作，不会导致TrollStore无法正常使用，比如变成User用户模式或打不开

所以需要去：

- 把TrollStore设置为持续存在 `Persistence`
  - 核心思路：找个（自己平时不用的）系统app 或 （比如此处）就用上面的 `GTA Car Tracker` ，去设置为 `Persistence Helper`
  - 具体步骤：点击 `GTA Car Tracker`
    - -》 `Register Persistence Helper`
      -


16:29
TrollStore Helper
INFO
TrollStore
Installed, 1.4.5
Refresh App Registrations
Uninstall TrollStore
Register Persistence Helper
If you want to use this app as the TrollStore persistence helper, you can register it here.
Powered by Fugu15 CoreTrust & installd bugs, thanks to @LinusHenze
© 2022 Lars Fröder (opa334)

- 注册后的效果是
  -

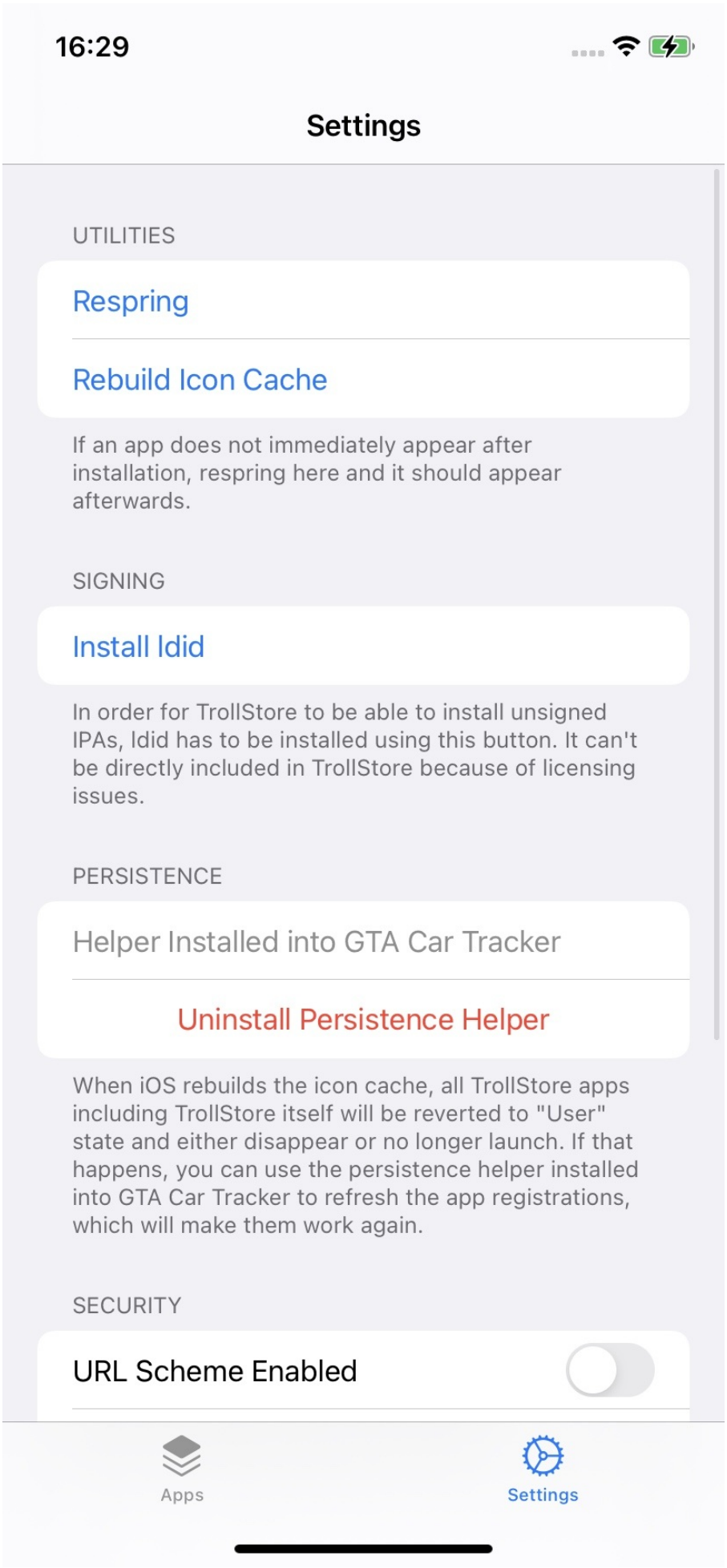

16:29
TrollStore Helper
INFO
TrollStore
Installed, 1.4.5
Refresh App Registrations
Uninstall TrollStore
Unregister Persistence Helper
This app is registered as the TrollStore persistence helper and can be used to fix TrollStore app registrations in case they revert back to "User" state and the apps say they're unavailable.
Powered by Fugu15 CoreTrust & installd bugs, thanks to @LinusHenze
© 2022 Lars Fröder (opa334)

- 详细过程参考官网文档：
  - TrollStore/install_trollhelperota_ios15.md at main · opa334/TrollStore · GitHub
- 注意：
  - 后续不能删除 `TrollStore Helper` ==此处的 `GTA Car Tracker`
    - 因为：上面通过GTA Car Tracker == TrollStore Helper，点击了其中的：Register Persistence Helper，意思是把GTA Car Tracker作为了一个系统的app，用于后续TrollStore的永久保持的功能，所以以后不能删除此app：GTA Car Tracker

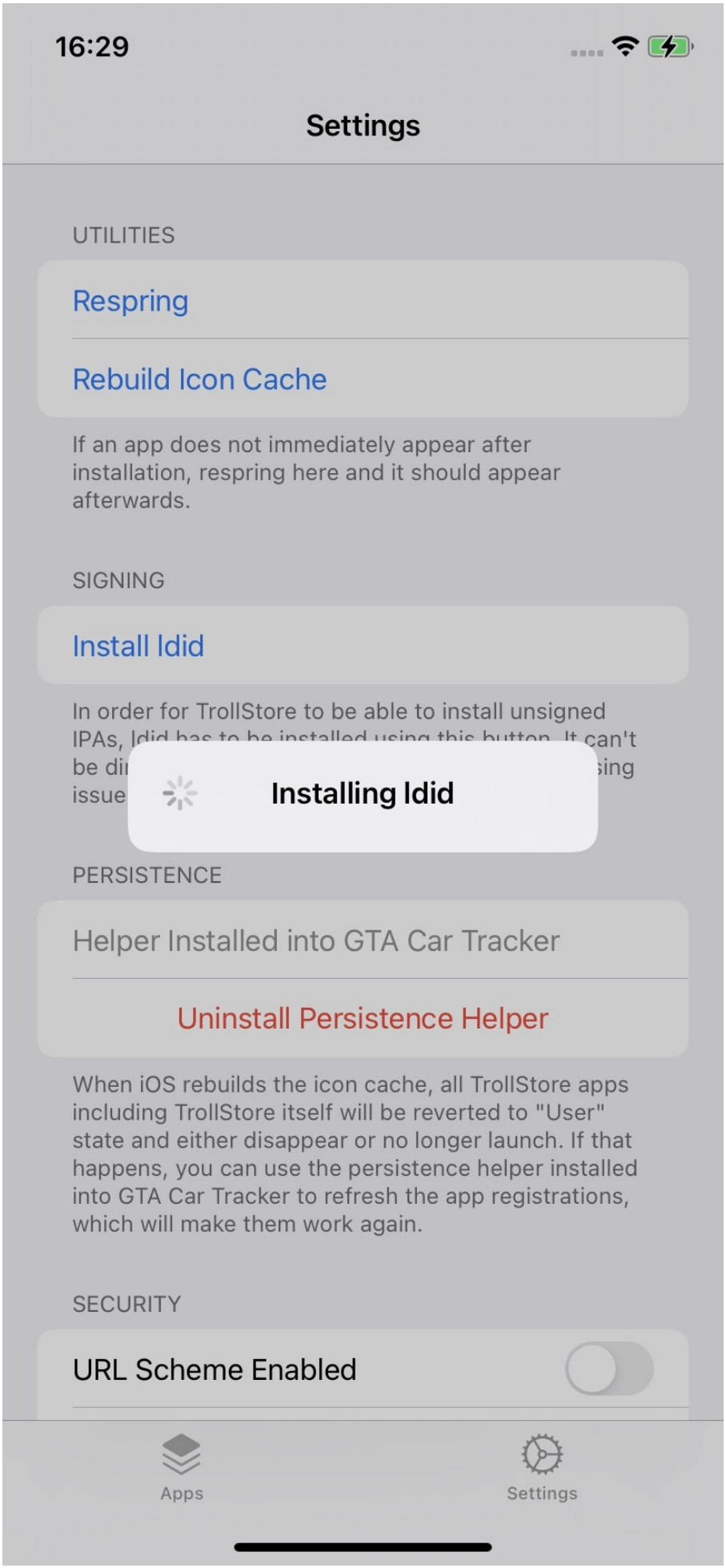
初始化配置TrollStore

安装完毕TrollStore后，还需要：

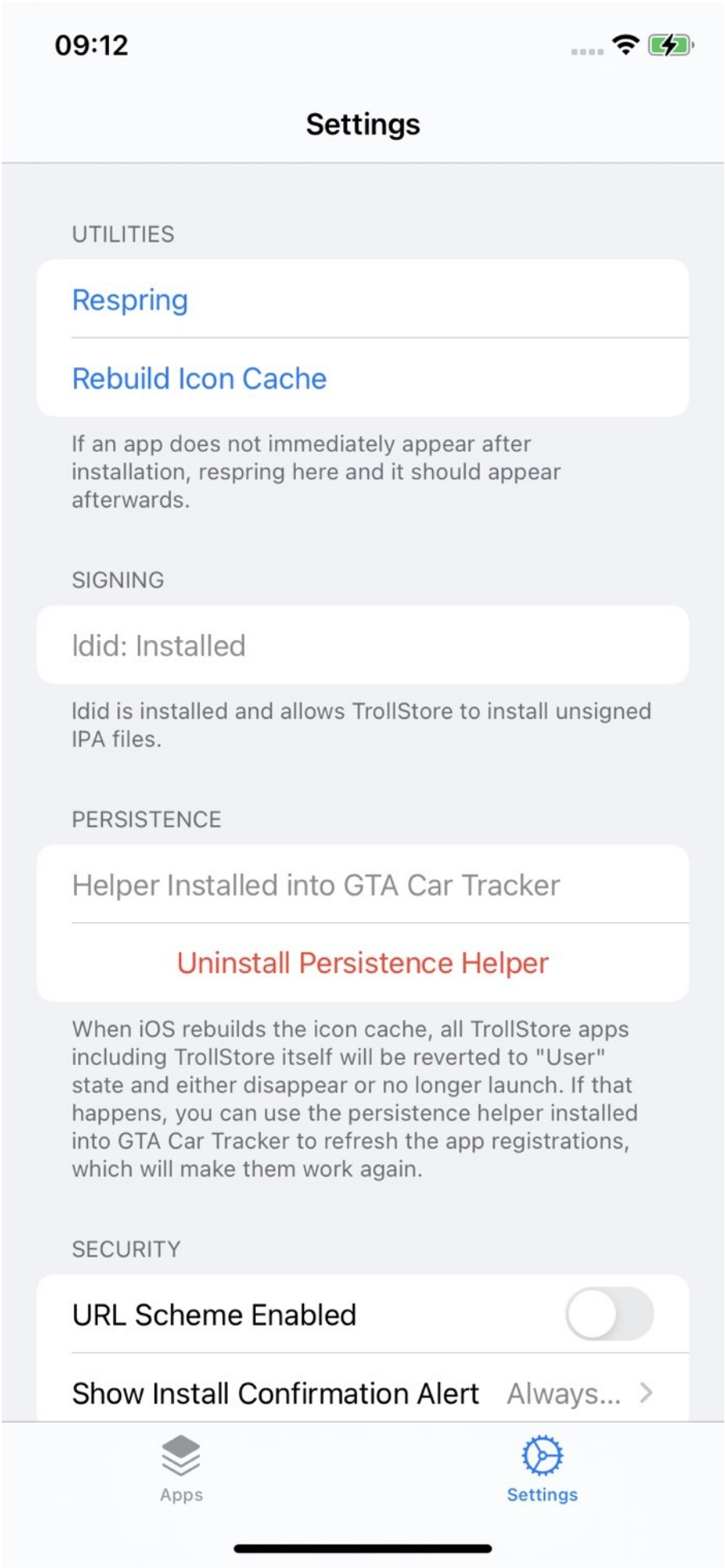
- 初始化配置TrollStore
  - 核心步骤：TrollStore -> Settings -> Install ldid
  - 详细步骤
    - TrollStore-> Settings -> Install ldid
      -


16:29
Settings
UTILITIES
Respring
Rebuild Icon Cache
If an app does not immediately appear after installation, respring here and it should appear afterwards.
SIGNING
Install ldid
In order for TrollStore to be able to install unsigned IPAs, ldid has to be installed using this button. It can't be directly included in TrollStore because of licensing issues.
PERSISTENCE
Helper Installed into GTA Car Tracker
Uninstall Persistence Helper
When iOS rebuilds the icon cache, all TrollStore apps including TrollStore itself will be reverted to "User" state and either disappear or no longer launch. If that happens, you can use the persistence helper installed into GTA Car Tracker to refresh the app registrations, which will make them work again.
SECURITY
URL Scheme Enabled
Apps
Settings

- 显示正在安装：`Installing ldid`
  -


16:29
Settings
UTILITIES
Respring
Rebuild Icon Cache
If an app does not immediately appear after installation, respring here and it should appear afterwards.
SIGNING
Install ldid
In order for TrollStore to be able to install unsigned IPAs, ldid has to be installed using this button. It can't be di sing issue
Installing ldid
PERSISTENCE
Helper Installed into GTA Car Tracker
Uninstall Persistence Helper
When iOS rebuilds the icon cache, all TrollStore apps including TrollStore itself will be reverted to "User" state and either disappear or no longer launch. If that happens, you can use the persistence helper installed into GTA Car Tracker to refresh the app registrations, which will make them work again.
SECURITY
URL Scheme Enabled
Apps
Settings

- ldid安装完毕后
  -


09:12
Settings
UTILITIES
Respring
Rebuild Icon Cache
If an app does not immediately appear after installation, respring here and it should appear afterwards.
SIGNING
ldid: Installed
ldid is installed and allows TrollStore to install unsigned IPA files.
PERSISTENCE
Helper Installed into GTA Car Tracker
Uninstall Persistence Helper
When iOS rebuilds the icon cache, all TrollStore apps including TrollStore itself will be reverted to "User" state and either disappear or no longer launch. If that happens, you can use the persistence helper installed into GTA Car Tracker to refresh the app registrations, which will make them work again.
SECURITY
URL Scheme Enabled
Show Install Confirmation Alert
Always...
Apps
Settings

- 会看到文字提示：`ldid is installed and allows TrollStore to install unsigned IPA files`

常见错误

Error downloading ldid Code 1001 请求超时

如果 `installing ldid` 期间：

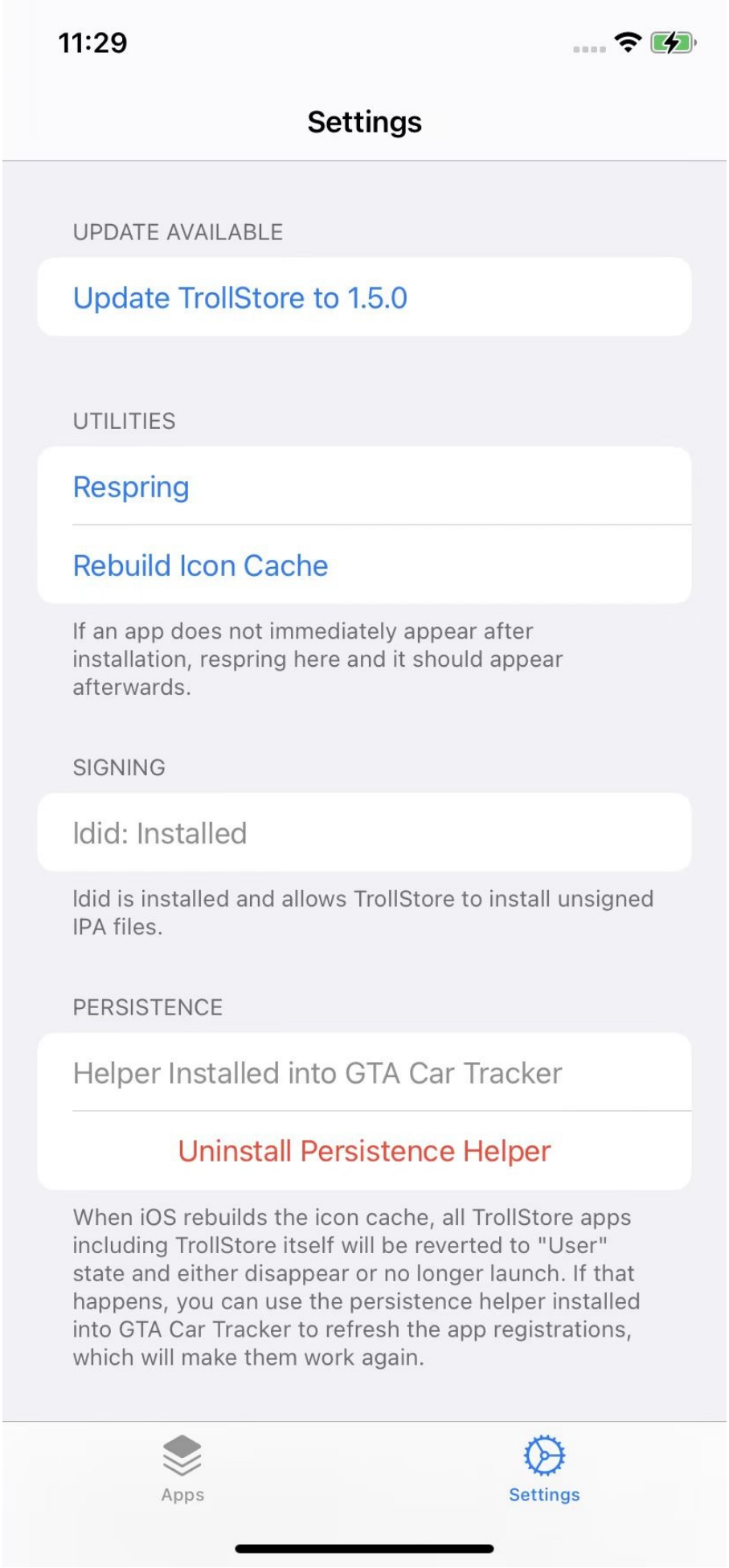
- 报错：Error downloading ldid Code 1001 请求超时
  -


16:30
Settings
UTILITIES
Error
Error downloading ldid: Error Domain=NSURLErrorDomain Code=-1001 "请求超时。" UserInfo={_kCFStreamErrorCodeKey=-2102, NSUnderlyingError=0x281e55ec0 {Error Domain=kCFErrorDomainCFNetwork Code=-1001 "(null)" UserInfo={_kCFStreamErrorCodeKey=-2102, _kCFStreamErrorDomainKey=4}}, _NSURLErrorFailingURLSessionTaskErrorKey=LocalDownloadTask <55DCDA91-DF7B-4F91-8881-7C8F74864595>.<2>, _NSURLErrorRelatedURLSessionTaskErrorKey=("LocalDownloadTask <55DCDA91-DF7B-4F91-8881-7C8F74864595>.<2>"), NSLocalizedDescription=请求超时。, NSErrorFailingURLStringKey=https://github.com/opa334/ldid/releases/download/v2.1.5-procursus5/ldid, NSErrorFailingURLKey=https://github.com/opa334/ldid/releases/download/v2.1.5-procursus5/ldid, _kCFStreamErrorDomainKey=4}
Close
URL Scheme Enabled
Apps
Settings

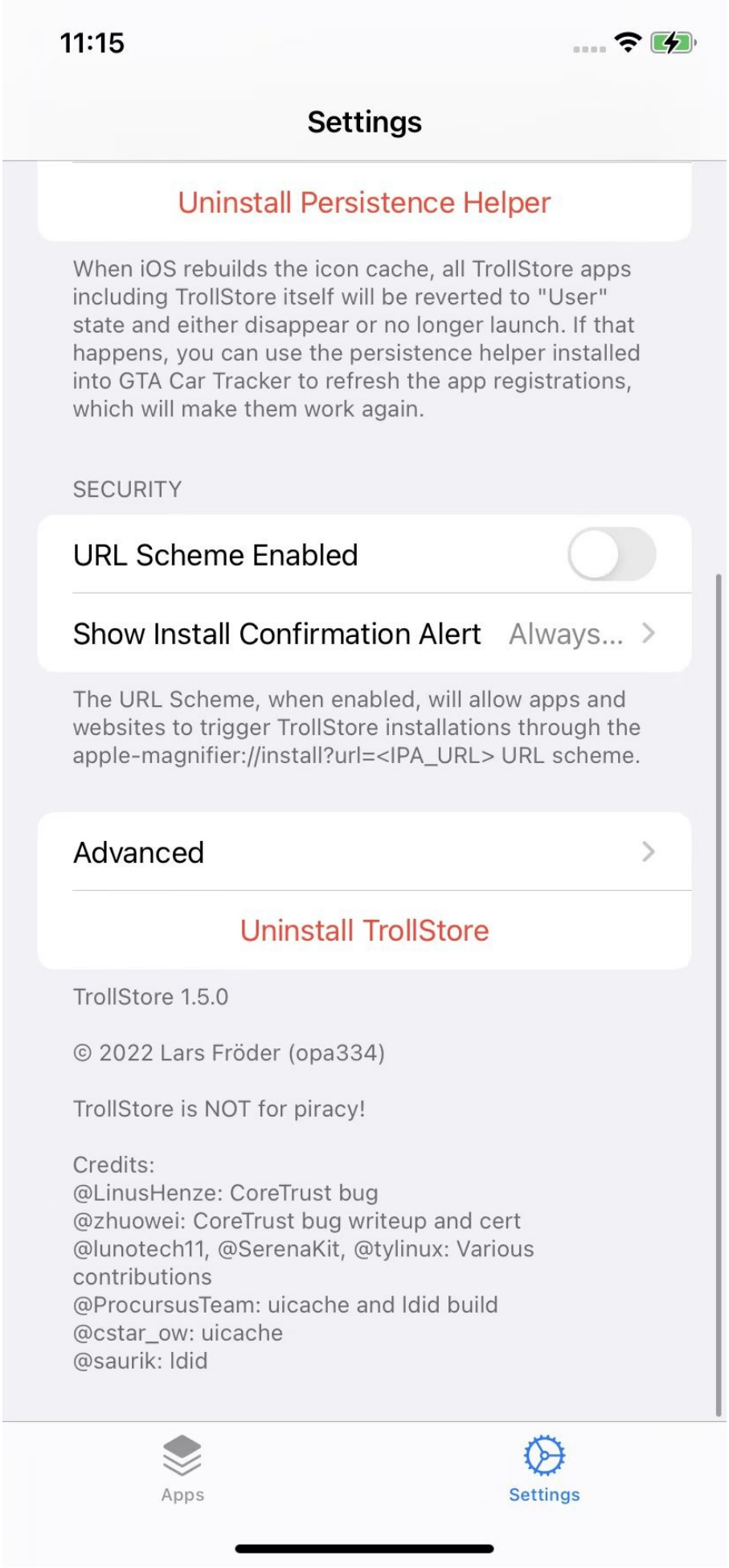
- 原因：无法访问外网（https://github.com/xxx）
- 解决办法：用Shadowrocket小火箭，加上代理，确保翻墙后可以正常上外网
  - 详见
    - 【已解决】给iOS 15.1的iPhone 11去翻墙科学上网安装代理

升级TrollStore

- 升级TrollStore
  - TrollStore中如果有新版本，则会有对应新版本提示
    - 此处的：Update TrollStore to 1.5.0
      -

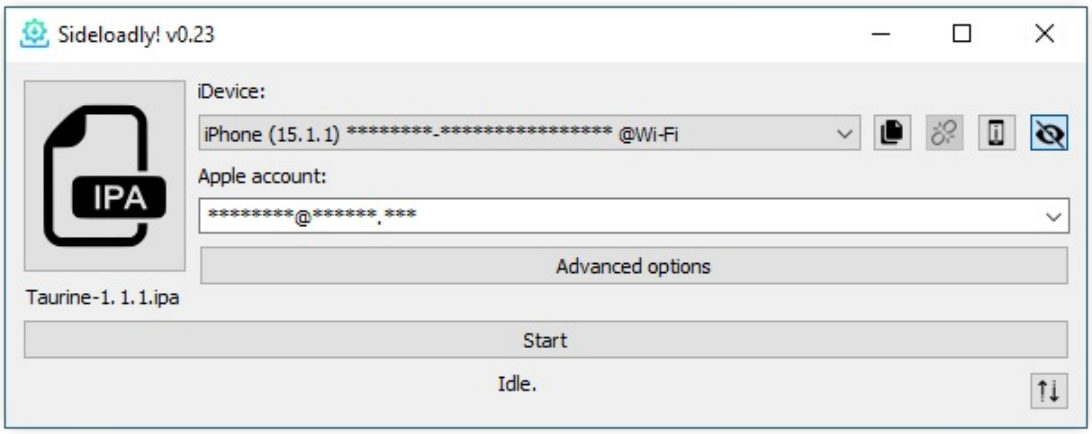

11:29
Settings
UPDATE AVAILABLE
Update TrollStore to 1.5.0
UTILITIES
Respring
Rebuild Icon Cache
If an app does not immediately appear after installation, respring here and it should appear afterwards.
SIGNING
ldid: Installed
ldid is installed and allows TrollStore to install unsigned IPA files.
PERSISTENCE
Helper Installed into GTA Car Tracker
Uninstall Persistence Helper
When iOS rebuilds the icon cache, all TrollStore apps including TrollStore itself will be reverted to "User" state and either disappear or no longer launch. If that happens, you can use the persistence helper installed into GTA Car Tracker to refresh the app registrations, which will make them work again.
Apps
Settings

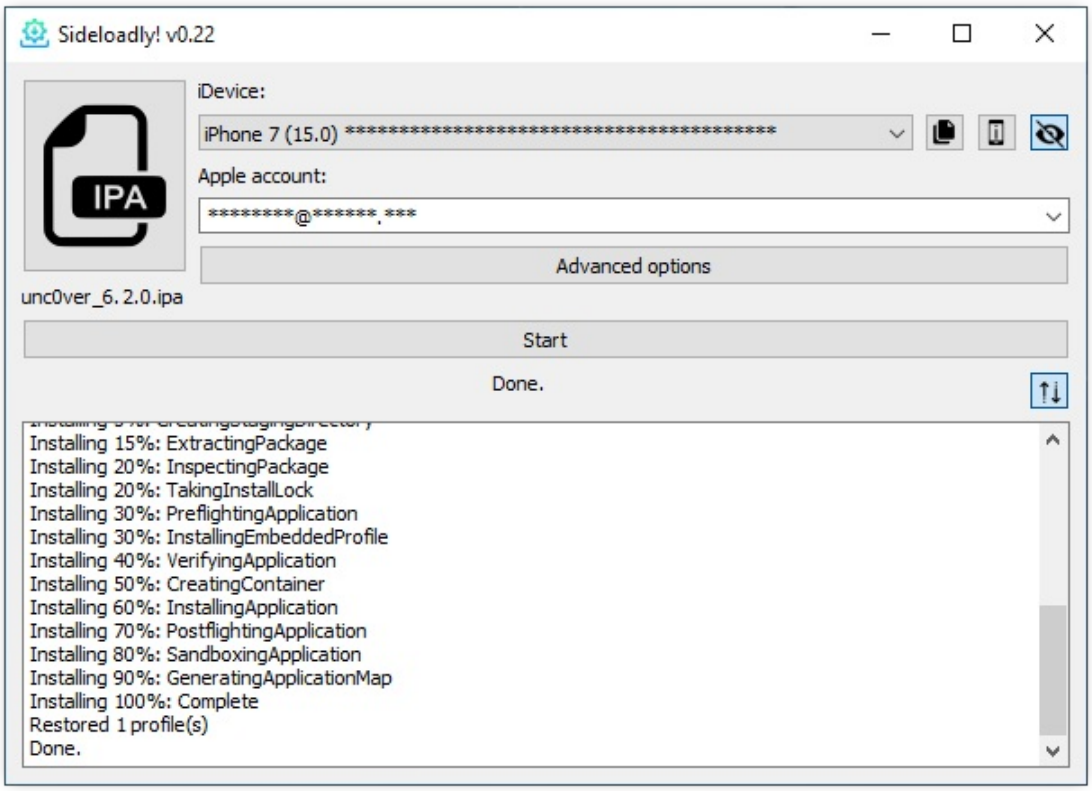
- 点击继续安装即可
  - 注：同理，确保能上外网，否则会出现下载失败的情况
- 更新后：1.5.0
  -

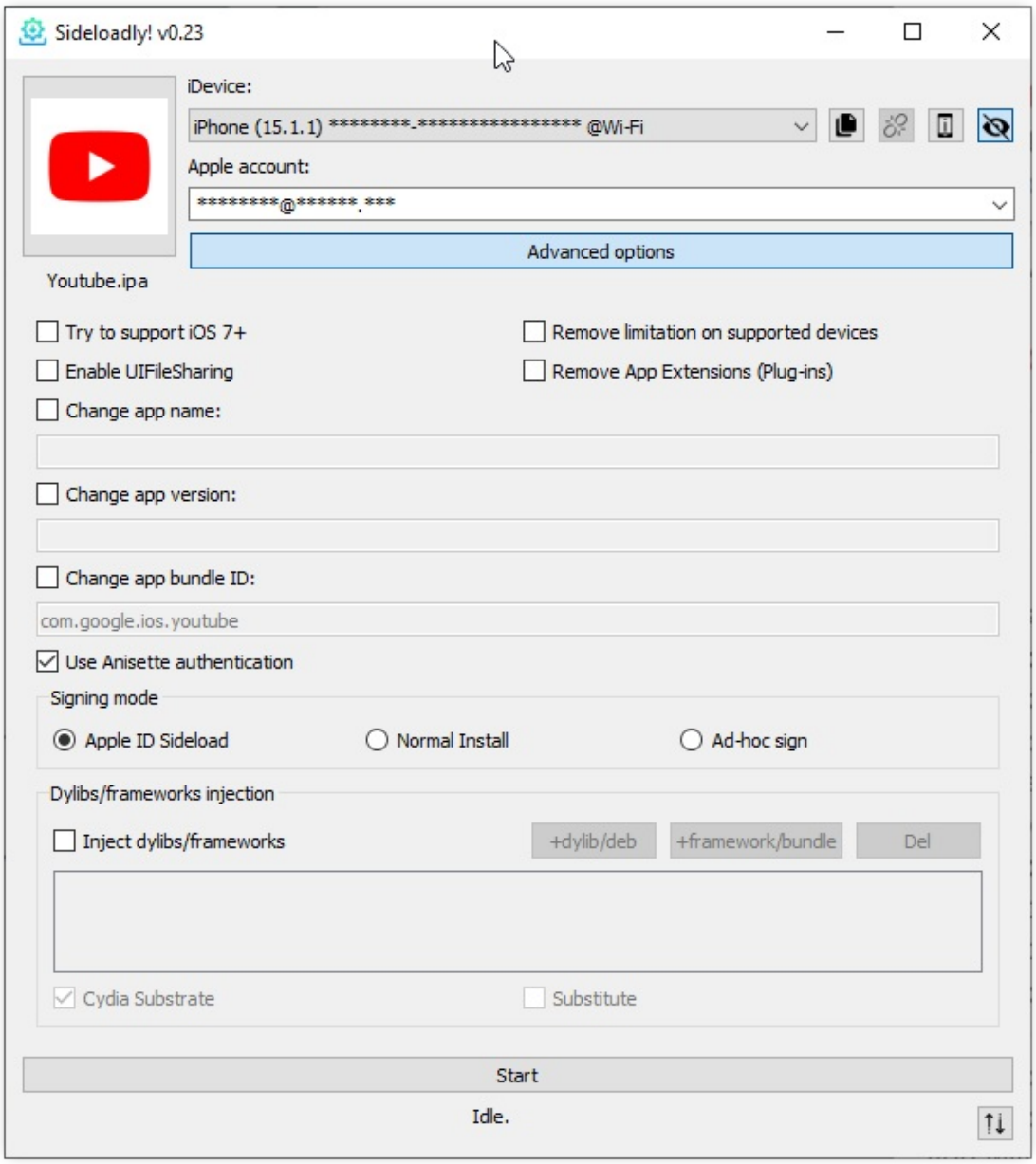

11:15
Settings
Uninstall Persistence Helper
When iOS rebuilds the icon cache, all TrollStore apps including TrollStore itself will be reverted to "User" state and either disappear or no longer launch. If that happens, you can use the persistence helper installed into GTA Car Tracker to refresh the app registrations, which will make them work again.
SECURITY
URL Scheme Enabled
Show Install Confirmation Alert
Always...
The URL Scheme, when enabled, will allow apps and websites to trigger TrollStore installations through the apple-magnifier://install?url=<IPA_URL> URL scheme.
Advanced
Uninstall TrollStore
TrollStore 1.5.0
© 2022 Lars Fröder (opa334)
TrollStore is NOT for piracy!
Credits:
@LinusHenze: CoreTrust bug
@zhuowei: CoreTrust bug writeup and cert
@lunotech11, @SerenaKit, @tylinux: Various contributions
@ProcursusTeam: uicache and ldid build
@cstar_ow: uicache
@saurik: ldid
Apps
Settings

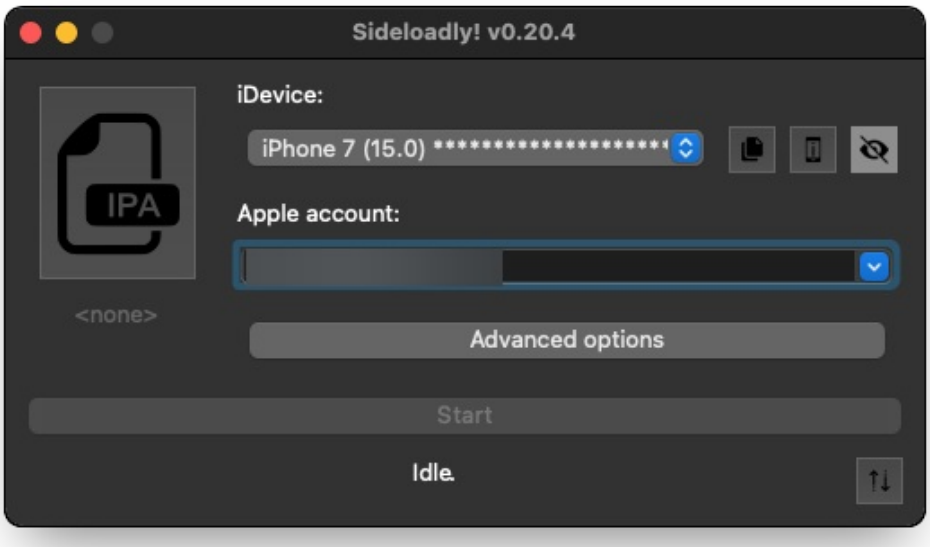
Sideloadly

- `Sideloadly`
  - 官网
    - https://sideloadly.io/
      - Sideloadly - iOS, Apple Silicon & TV Sideloading
    - https://sideloadly.app/
      - Sideloadly - Permanently Sideload IPA files FREE (Installer)
  - 下载地址
    - Mac
      - https://sideloadly.app/SideloadlySetup.dmg
    - Win
      - 64bit
        - https://sideloadly.app/SideloadlySetup64.exe
      - 32bit
        - https://sideloadly.app/SideloadlySetup32.exe
  - 截图
    - 

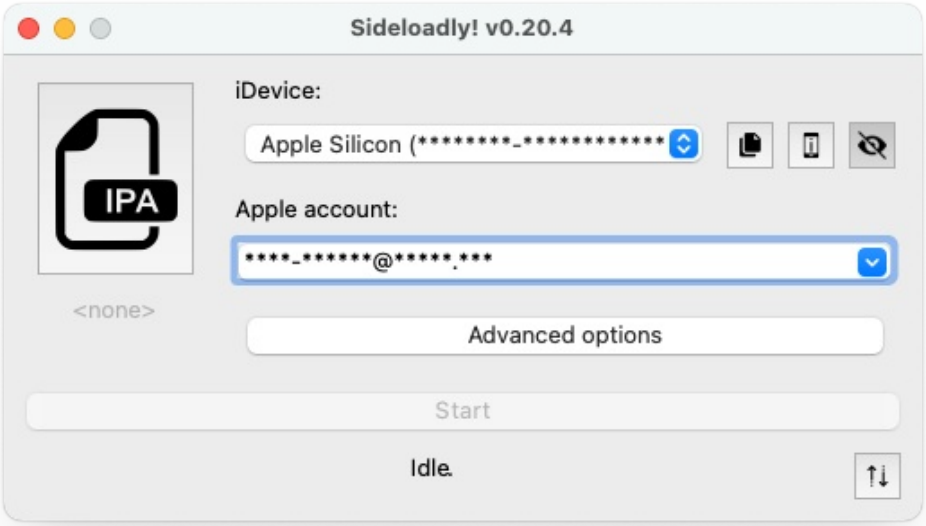
- 
Sideloadly! v0.23
IPA
iDevice:
iPhone (15.1.1) ********-********** @Wi-Fi
Apple account:
********@******.***
Advanced options
Taurine-1.1.1.ipa
Start
Idle.

- 
Sideloadly! v0.22
IPA
iDevice:
iPhone 7 (15.0) **
Apple account:
********@******.***
Advanced options
unc0ver_6.2.0.ipa
Start
Done.
Installing 15%: ExtractingPackage
Installing 20%: InspectingPackage
Installing 20%: TakingInstallLock
Installing 30%: PreflightingApplication
Installing 30%: InstallingEmbeddedProfile
Installing 40%: VerifyingApplication
Installing 50%: CreatingContainer
Installing 60%: InstallingApplication
Installing 70%: PostflightingApplication
Installing 80%: SandboxingApplication
Installing 90%: GeneratingApplicationMap
Installing 100%: Complete
Restored 1 profile(s)
Done.

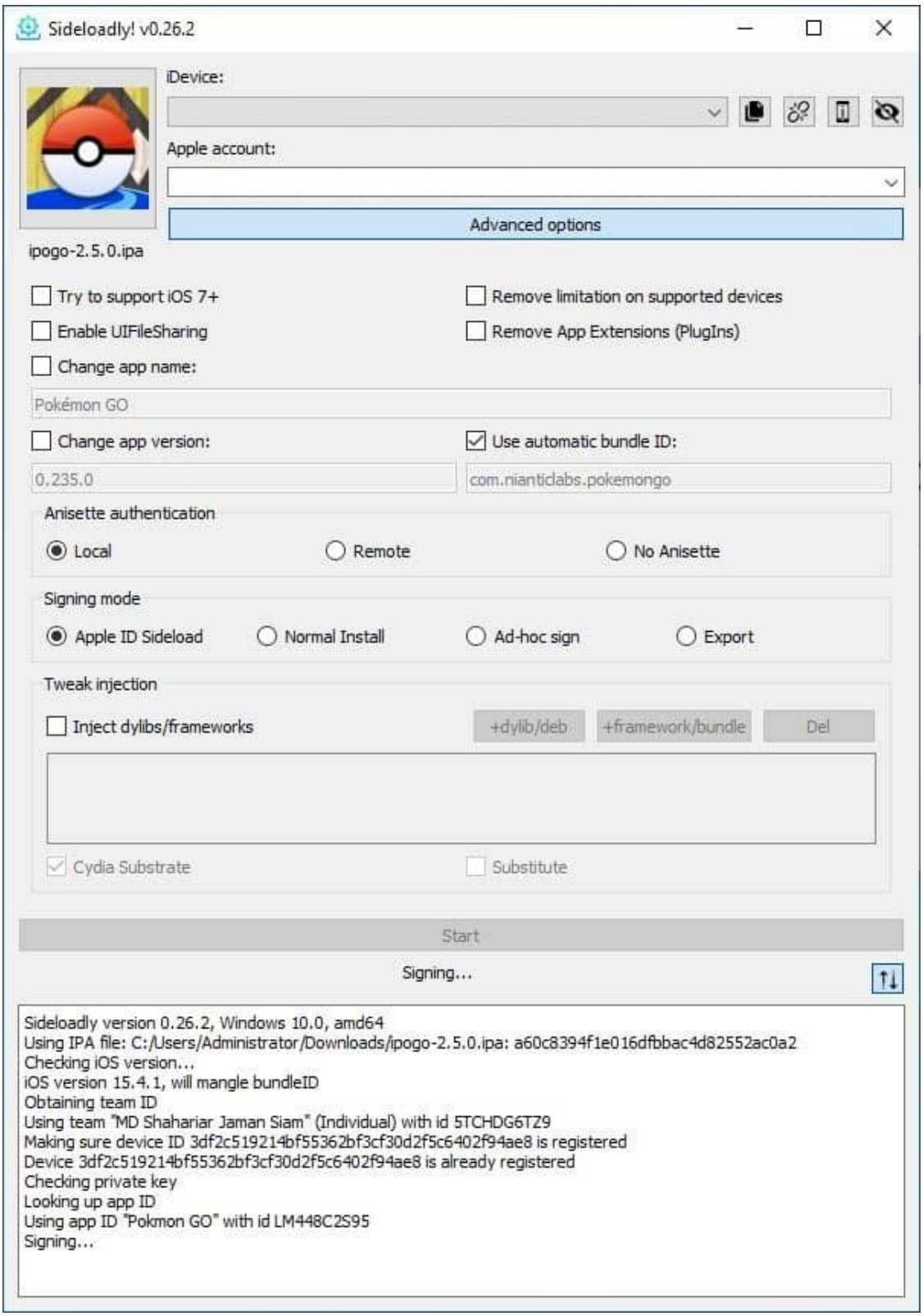

Sideloadly! v0.23
iDevice:
iPhone (15.1.1) ********_********** @Wi-Fi
Apple account:
********@******.***
Advanced options
Youtube.ipa
Try to support iOS 7+
Remove limitation on supported devices
Enable UIFileSharing
Remove App Extensions (Plug-ins)
Change app name:
Change app version:
Change app bundle ID:
com.google.ios.youtube
Use Anisette authentication
Signing mode
Apple ID Sideload
Normal Install
Ad-hoc sign
Dylibs/frameworks injection
Inject dylibs/frameworks
+dylib/deb
+framework/bundle
Del
Cydia Substrate
Substitute
Start
Idle.

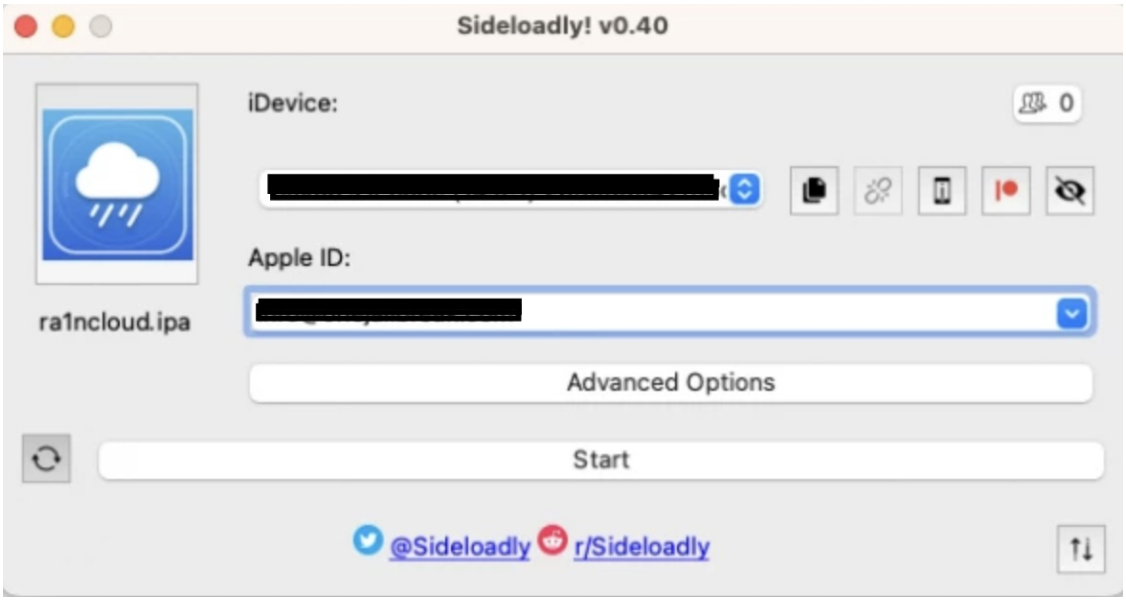

Sideloadly! v0.20.4
iDevice:
iPhone 7 (15.0) ********
IPA
Apple account:
<none>
Advanced options
Start
Idle.

-


Sideloadly! v0.20.4
iDevice:
Apple Silicon (********-**********
IPA
Apple account:
****_******@*****.***
<none>
Advanced options
Start
Idle.

-


Sideloadly! v0.26.2
iDevice:
Apple account:
Advanced options
ipogo-2.5.0.ipa
Try to support iOS 7+
Remove limitation on supported devices
Enable UIFileSharing
Remove App Extensions (PlugIns)
Change app name:
Pokémon GO
Change app version:
Use automatic bundle ID:
0.235.0
com.nianticlabs.pokemongo
Anisette authentication
Local
Remote
No Anisette
Signing mode
Apple ID Sideload
Normal Install
Ad-hoc sign
Export
Tweak injection
Inject dylibs/frameworks
+dylib/deb
+framework/bundle
Del
Cydia Substrate
Substitute
Start
Signing...
Sideloadly version 0.26.2, Windows 10.0, amd64
Using IPA file: C:/Users/Administrator/Downloads/ipogo-2.5.0.ipa: a60c8394f1e016dfbbac4d82552ac0a2
Checking iOS version...
iOS version 15.4.1, will mangle bundleID
Obtaining team ID
Using team "MD Shahariar Jaman Siam" (Individual) with id 5TCHDG6TZ9
Making sure device ID 3df2c519214bf55362bf3cf30d2f5c6402f94ae8 is registered
Device 3df2c519214bf55362bf3cf30d2f5c6402f94ae8 is already registered
Checking private key
Looking up app ID
Using app ID "Pokmon GO" with id LM448C2S95
Signing...



palera1n

- palera1n
  - 资料
    - 主页
      - https://palera.in/
        - palera1n is a developer-oriented jailbreak for checkm8 devices (A8-A11) on iOS 15.0-16.3
    - github
      - https://github.com/palera1n/
        - https://github.com/palera1n/palera1n
  - 前提条件
    - checkm8 vulnerable iOS device
  - 支持的iOS版本： `iOS/iPadOS 15+`
    - iOS 15.x or 16.x
      - iOS 15.0 ~ iOS 16.3.1
  - 支持的iOS设备
    - 芯片类型/架构： `arm64`
      - `A11` 及之前
        - 注： `A12+` （架构是 `arm64e` ）就不支持了
    - 机型
      - iPhone 6s Plus
      - iPhone SE (2016)
      - iPhone 7
      - iPhone 7 Plus
      - iPhone 8
      - iPhone 8 Plus
      - iPhone X
      -
      - iPad mini 4
      - iPad Air 2
      - iPad (5th generation)
      - iPad (6th generation)
      - iPad (7th generation)
      - iPad Pro (9.7")
      - iPad Pro (12.9") (1st generation)
      - iPad Pro (10.5")
      - iPad Pro (12.9") (2nd generation)
      -
      - iPod Touch (7th generation)
  - 支持2种越狱模式/越狱类型
    - `rootful` = `fakefs-rootful` = `普通越狱` = `有根越狱` ： rootfs可写，包括根目录/也可以写
    - `rootless` = `无根越狱` ： rootfs只读，只有/var可写
  - 推荐的越狱模式
    - rootful越狱=有根越狱=普通越狱
      - 这样对于之前的兼容性会更好
      - 很多插件等，应该可以正常工作了
      - 比如希望的 `frida` 等等

palera1n越狱前

palera1n越狱的前提条件

- 要满足一系列条件
  - If you want the device to be semi-tethered, you will need 5-10GB of space for the fakefs. This means that 16GB devices cannot be semi-tethered
  - If you are on A10(X), use checkp4le instead for full SEP functionality (Passcode, TouchID, Apple Pay)
  - On A11, you must disable your passcode while in the jailbroken state (on iOS 16, you need to reset your device before proceeding with palera1n A11).
  - USB-A cables are recommended to use, USB-C may have issues with palera1n and getting into DFU mode.
  - A Linux or macOS computer
    - Python 3 must be installed
    - AMD CPUs have an issue [with (likely) their USB controllers] that causes them to have a very low success rate with checkm8. It is not recommended that you use them with palera1n. If your device does not successfully jailbreak, try a computer with an Intel or other CPU

palera1n越狱的注意事项和说明

- 注意事项和说明
  - palera1n jailbreaks any iOS/iPadOS device with an arm64 (arm64e excluded) on iOS 15+, utilizing the checkm8 bootROM exploit.
    - 不支持A11之后的arm64e
      - 注：我之前的iPhone11，就是A12芯片，就是arm64e，所以不支持
  - arm64e devices will NEVER be supported.
    - 永远不会支持arm64e
  - palera1n is able to jailbreak the device in fakefs-rootful mode, where / is writable, as well as rootless mode, where / cannot be written to.
    - palera1n支持：
      - fakefs-rootful = rootful=伪造根文件系统 越狱：根目录/可写入
      - rootless=无根越狱：根目录/不可写入
  - Due to the nature of the checkm8 exploit, palera1n is semi-tethered. That is, you must run the palera1n tool after the device reboot in order to enter the jailbroken state. However, it is not required for the device to boot.
    - 是非完美越狱：
      - 原因：checkm8决定的
        - 结果：每次重启（iPhone）后，要重新运行palera1n（去恢复越狱）
          - 注：启动boot时不需要
  - On A11 devices, that is, iPhone 8, iPhone 8 Plus and iPhone X, the passcode cannot be used.
    - A11设备（iPhone8、iPhone 8P、iPhoneX）中，不能用passcode
  - On iOS 15, the passcode must be off while jailbroken.
    - iOS 15中必须关闭passcode（才能越狱）
  - On iOS 16, the passcode must be off since restore, and Reset All Contents and Settings from settings app counts as a restore. A backup may be used in this case.
    - iOS 16中，passcode必须关闭，且如果之前开启过passcode，则需要恢复出厂设置=重置系统

要清楚palera1n越狱工具的版本

- `palera1n` 越狱工具的版本
  - Windows用：`palen1x`
    - 文档：
      - Using palen1x | iOS Guide (cfw.guide)
      - palera1n/docs: GitBook docs for palera1n (github.com)
        - docs/flashing-palen1x.md
        - docs/booting-palen1x.md
        - docs/jailbreak-with-palen1x.md
    - 代码仓库
      - palera1n/palen1x: Alpine-based distro that lets you install rootful and rootless palera1n-c. (github.com)
        - https://github.com/palera1n/palen1x/
  - Linux/Mac的用：`palera1n`，有2个版本
    - `shell script`
      - 旧版本是个 `palera1n.sh` 脚本
      - 旧版本的教程
        - Installing palera1n (Legacy) | iOS Guide (cfw.guide)
          - 核心命令
            - `./palera1n.sh --tweaks 15.6.1 --semi-tethered`
    - （编译好的 电脑端的 直接可用的） `二进制`
      - 代码仓库
        - palera1n/palera1n-c: palera1n written in C (github.com)
          - palera1n written in C
            - 是个用C写的，在PC端（Mac、Linux等）中运行的，palera1n的二进制程序
      - 下载地址
        - Releases · palera1n/palera1n-c (github.com)
          - macOS
            - palera1n-macos-universal
              - 截至20230302最新版：v2.0.0-beta.4
                - https://github.com/palera1n/palera1n-c/releases/download/v2.0.0-beta.4/palera1n-macos-universal
      - 常用参数
        - `-c`，`--setup-fakefs` Setup fakefs
          - When used with -f, --fakefs, Create the new APFS volume required for rootful. Will fail if one already exists.
        - `-f`，`--fakefs` Boots fakefs
          - Jailbreak in rootful mode.
        - `-l`，`--rootless` Boots rootless. This is the default
        - `-v`，`--debug-logging` Enable debug logging
          - This option can be repeated for extra verbosity.
        - `-V`，`--verbose-boot` Verbose boot
      - 常见用法
        - `palera1n -cf` == `palera1n -c -f`
          - 创建fakefs
        - `palera1n -f`
          - 启动设备
        - `palera1n` == `palera1n -l` == `palera1n --rootless`
          - 无任何参数的启动，（默认）以rootless方式去越狱
            - 注意：rootless模式下支持的tweak插件很少
        - 常见用法对比

        - `palera1n -f`：`f`=`Fakefs`=`rootFul`
        - `palera1n`==`palera1n -l`：`l`=`rootLess`
    - 完整用法=语法=帮助
        - `palera1n --help`
        - 或 在线的html文档
            - palera1n - nickchan.lol
                - https://cdn.nickchan.lol/palera1n/c-rewrite/releases/v2.0.0-beta.4/palera1n.1.html
                - https://cdn.nickchan.lol/palera1n/artifacts/c-rewrite/palera1n.1.html

palera1n的文档和资料

- palera1n/palera1n: iOS 15.0-16.3 (semi-)tethered checkm8 jailbreak (github.com)
- palera1n(1) (nickchan.lol)
-
- Installing palera1n | iOS Guide (cfw.guide)
- palera1n/COMMONISSUES.md at main · palera1n/palera1n (github.com)
- palera1n/CHANGELOG.md at main · palera1n/palera1n (github.com)
-
- palera1n/docs: GitBook docs for palera1n (github.com)
- docs/run-on-macos.md at main · palera1n/docs (github.com)
- docs/jailbreak.md at main · palera1n/docs (github.com)
-
- docs/flashing-palen1x.md at main · palera1n/docs (github.com)
- docs/booting-palen1x.md at main · palera1n/docs (github.com)
- docs/jailbreak-with-palen1x.md at main · palera1n/docs (github.com)

palera1n越狱过程

此处介绍：用palera1n给iOS 15.0的iPhone8越狱

- 待越狱设备
  - `iOS 15.0`，`iPhone 8`（`arm64` 的 `A11`）
    - 

  - 待越狱手机，已满足相关前提条件：
    - 是256GB，满足rootful越狱对空间的要求：5~10G空闲空间
    - iPhone手机的芯片是A11，是arm64
    - A11的iPhone8中已禁用passcode
    - UBS数据线是USB-A
    - 电脑是Mac（MacOS）
      - 是Intel的CPU
      - 已安装过Python3
- 此处越狱模式选择：`rootful jailbreak` = `普通越狱` = `有根越狱`

用palera1n给iOS 15.0的iPhone8越狱过程概述

- palera1n越狱的核心步骤
  - Mac中给iPhone越狱
    - Mac中
      - 下载Mac版的 `palera1n-macos-universal`
        - 此处版本：palera1n v2.0.0-beta.4
      - `palera1n -c -f`
        - Enter
        - 进入DFU模式
          - 长按 `音量减键` 和 `电源键`
          - （不要松手，继续）长按 `音量减键`
      - `palera1n -f`
    - iPhone中
      - palera1n的app中：点击 `Install`

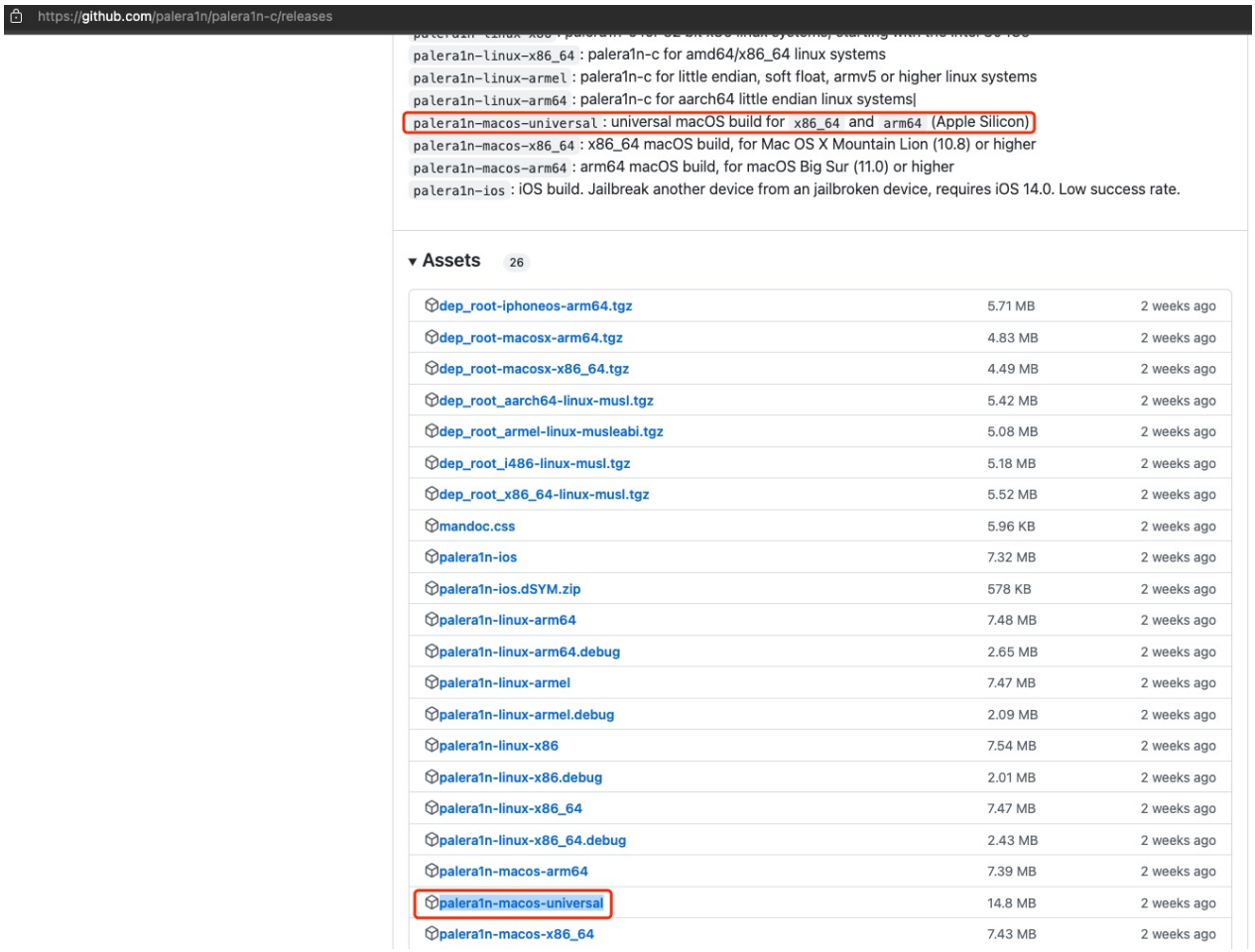
用palera1n给iOS 15.0的iPhone8越狱的详细过程

第一步：下载palera1n的二进制

此处下载：Mac的palera1n的二进制文件：

palera1n-macos-universal

https://github.com/palera1n/palera1n-c/releases/download/v2.0.0-beta.4/palera1n-macos-universal



并放到合适的目录中，比如：

`/usr/local/bin/palera1n`

此过程：

- 可以手动操作
- 也可以用命令去操作

```
sudo curl -Lo /usr/local/bin/palera1n https://github.com/palera1n/palera1n-c/releases/download/v2.0.0-beta.4/palera1n-macos-universal
sudo chmod +x /usr/local/bin/palera1n
```

或：

```
sudo mv ./palera1n-macos-universal /usr/local/bin/
mv /usr/local/bin/palera1n-macos-universal /usr/local/bin/palera1n
sudo xattr -c /usr/local/bin/palera1n
sudo chmod +x /usr/local/bin/palera1n
```

放好后，确保命令行可以找到：

```
> which palera1n
/usr/local/bin/palera1n
```

另外顺带去看看版本：

```
crifan@licrifandeMacBook-Pro ~/dev/dev_tool/reverse_security/iOS/palera1n palera1n --version
palera1n version 2.0.0: Wed Feb 15 08:49:44 UTC 2023; runner:v2.0.0-beta.4/RELEASE
# == palera1n-c ==
#
# Made by: Nick Chan, Ploosh, Mineek, Nebula, llsc12
#
# Thanks to: dora2ios, pythonplayer, tihmstar, nikias
# (libimobiledevice), checkra1n team (Siguza, axi0mx, littlelailo
# et al.), Procursus Team (Hayden Seay, Cameron Katri, Keto et.al)
```

第二步：palera1n -c -f，安装创建fakefs

然后就可以开始用palera1n去越狱了：

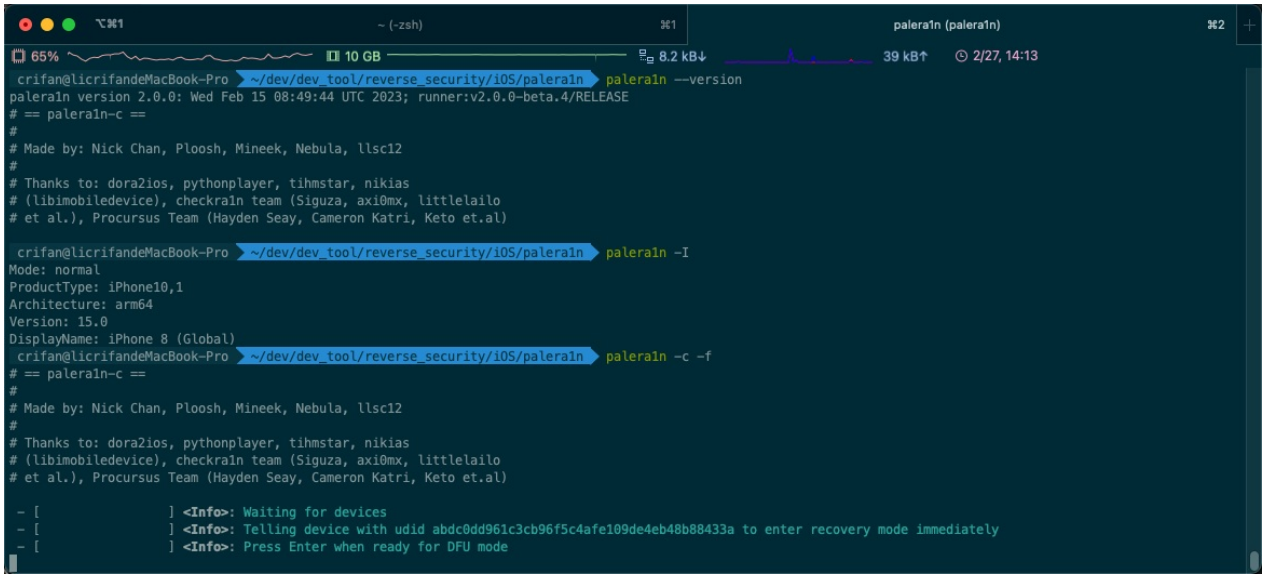
```
palera1n -c -f
```

其中：

- `-c`, `--setup-fakefs` Setup fakefs
  - When used with -f, --fakefs, Create the new APFS volume required for rootful. Will fail if one already exists.
    - 创建fakefs
- `-f`, `--fakefs` Boots fakefs
  - Jailbreak in rootful mode.
    - 越狱方式/类型/模式选择：普通越狱=rootful越狱

详细log日志：

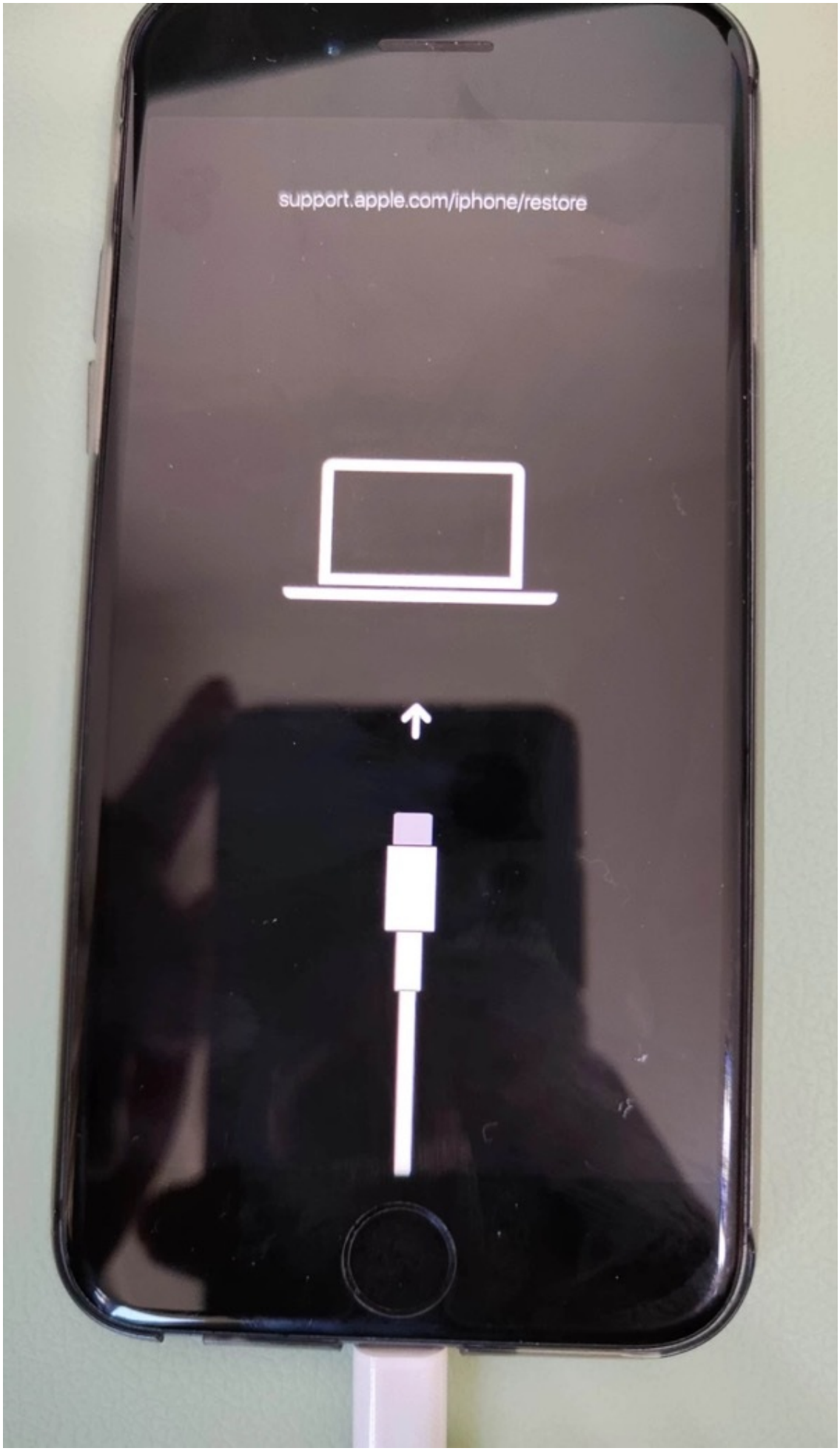
```
crifan@licrifandeMacBook-Pro ~/dev/dev_tool/reverse_security/iOS/palera1n palera1n -c -f
# == palera1n-c ==
#
# Made by: Nick Chan, Ploosh, Mineek, Nebula, llsc12
#
# Thanks to: dora2ios, pythonplayer, tihmstar, nikias
# (libimobiledevice), checkra1n team (Siguza, axi0mx, littlelailo
# et al.), Procursus Team (Hayden Seay, Cameron Katri, Keto et.al)

 - [02/27/23 14:13:08] Info : Waiting for devices
 - [02/27/23 14:13:08] Info : Telling device with udid abdc0dd961c3cb96f5c4afe109de4eb40b88433a to enter recovery mo
de immediately
 - [02/27/23 14:13:20] Info : Press Enter when ready for DFU mode
```



此时：iPhone手机中出现：

- 数据线插入电脑
  - 顶部文字：`support.apple.com/iphone/restore`

的界面：


support.apple.com/iphone/restore

然后去：

- Enter=回车

确认准备好，提示：`get ready`

再根据提示：

```
Get ready (0)
Hold volume down + side button (0)
Hold volume down button (3)
```

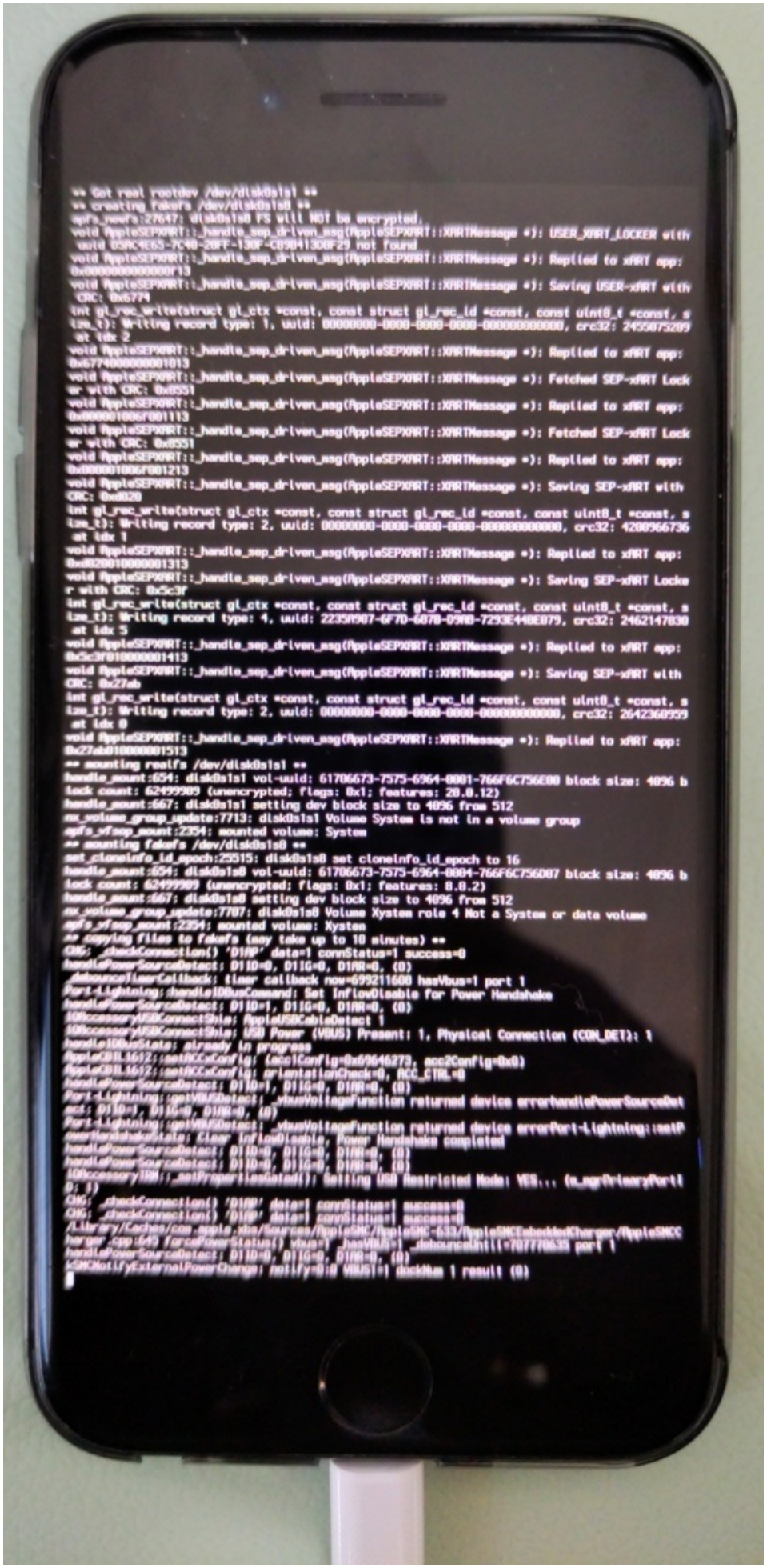
- 去操作iPhone进入DFU模式
  - Hold volume down + side button 长按 音量键减键 + 侧边栏键=电源键
  - Hold volume down button （保持不松手，继续）长按 音量键减键

即可继续，进入DFU模式，继续自动越狱过程

详细log日志：

```
- [02/27/23 14:37:54] Info : Device entered DFU mode successfully
- [02/27/23 14:37:54] Info : About to execute checkra1n
#
# Checkra1n 0.1337.1
#
# Proudly written in nano
# (c) 2019-2023 Kim Jong Cracks
#
#========= Made by ========
# argp, axi0mx, danyl931, jaywalker, kirb, littlelailo, nitoTV
# never_released, nullpixel, pimskeks, qwertyoruiop, sbingner, siguza
#========= Thanks to ========
# haifisch, jndok, jonseals, xerub, lilstevie, psychotea, sferrini
# Cellebrite (ih8sn0w, cjori, ronyrus et al.)
#============================

- [02/27/23 14:37:54] Verbose : Starting thread for Apple TV 4K Advanced board
- [02/27/23 14:37:54] Info : Waiting for DFU mode devices
- [02/27/23 14:37:54] Verbose : DFU mode device found
- [02/27/23 14:37:54] Info : Checking if device is ready
- [02/27/23 14:37:54] Verbose : Attempting to perform checkm8 on 8015 11
- [02/27/23 14:37:54] Info : Setting up the exploit
- [02/27/23 14:37:54] Verbose : == checkm8 setup stage ==
- [02/27/23 14:37:54] Verbose : Entered initial checkm8 state after 1 steps
- [02/27/23 14:37:54] Verbose : Stalled input endpoint after 4 steps
- [02/27/23 14:37:54] Verbose : DFU mode device disconnected
- [02/27/23 14:37:54] Verbose : DFU mode device found
- [02/27/23 14:37:54] Verbose : == checkm8 trigger stage ==
- [02/27/23 14:37:57] Info : Checkmate
- [02/27/23 14:37:57] Verbose : Device should now reconnect in download mode
- [02/27/23 14:37:57] Verbose : DFU mode device disconnected
- [02/27/23 14:38:04] Info : Entered download mode
- [02/27/23 14:38:04] Verbose : Download mode device found
- [02/27/23 14:38:04] Info : Booting PongoOS...
- [02/27/23 14:38:06] Info : Found PongoOS USB Device
- [02/27/23 14:38:06] Info : Booting Kernel...
crifan@licrifandeMacBook-Pro ~/dev/dev_tool/reverse_security/iOS/palera1n
```

然后手机上会输出很多log日志：


** Got real rootdev /dev/disk0s1s1 **
** creating fakefs /dev/disk0s1s0 **
apfs_newfs:27647: disk0s1s0 FS will NOT be encrypted.
void AppleSEPXART::_handle_sep_driven_msg(AppleSEPXART::XARTMessage *): USER_XART_LOCKER with
uuid 05AC4E65-7C40-20FF-130F-C050413D8F29 not found
void AppleSEPXART::_handle_sep_driven_msg(AppleSEPXART::XARTMessage *): Replied to xART app:
0x0000000000000f13
void AppleSEPXART::_handle_sep_driven_msg(AppleSEPXART::XARTMessage *): Saving USER-xART with
CRC: 0x6774
int gl_rec_write(struct gl_ctx *const, const struct gl_rec_id *const, const uint8_t *const, s
ize_t): Writing record type: 1, uuid: 00000000-0000-0000-0000-000000000000, crc32: 2455875209
at idx 2
void AppleSEPXART::_handle_sep_driven_msg(AppleSEPXART::XARTMessage *): Replied to xART app:
0x6774000000001013
void AppleSEPXART::_handle_sep_driven_msg(AppleSEPXART::XARTMessage *): Fetched SEP-xART Lock
er with CRC: 0x8551
void AppleSEPXART::_handle_sep_driven_msg(AppleSEPXART::XARTMessage *): Replied to xART app:
0x000001006f001113
void AppleSEPXART::_handle_sep_driven_msg(AppleSEPXART::XARTMessage *): Fetched SEP-xART Lock
er with CRC: 0x8551
void AppleSEPXART::_handle_sep_driven_msg(AppleSEPXART::XARTMessage *): Replied to xART app:
0x000001006f001213
void AppleSEPXART::_handle_sep_driven_msg(AppleSEPXART::XARTMessage *): Saving SEP-xART with
CRC: 0xd020
int gl_rec_write(struct gl_ctx *const, const struct gl_rec_id *const, const uint8_t *const, s
ize_t): Writing record type: 2, uuid: 00000000-0000-0000-0000-000000000000, crc32: 4200966736
at idx 1
void AppleSEPXART::_handle_sep_driven_msg(AppleSEPXART::XARTMessage *): Replied to xART app:
0xd02001000001313
void AppleSEPXART::_handle_sep_driven_msg(AppleSEPXART::XARTMessage *): Saving SEP-xART Locke
r with CRC: 0x5c3f
int gl_rec_write(struct gl_ctx *const, const struct gl_rec_id *const, const uint8_t *const, s
ize_t): Writing record type: 4, uuid: 2235F907-6F7D-607B-D90D-7293E448E879, crc32: 2462147830
at idx 5
void AppleSEPXART::_handle_sep_driven_msg(AppleSEPXART::XARTMessage *): Replied to xART app:
0x5c3f010000001413
void AppleSEPXART::_handle_sep_driven_msg(AppleSEPXART::XARTMessage *): Saving SEP-xART with
CRC: 0x27ab
int gl_rec_write(struct gl_ctx *const, const struct gl_rec_id *const, const uint8_t *const, s
ize_t): Writing record type: 2, uuid: 00000000-0000-0000-0000-000000000000, crc32: 2642360999
at idx 0
void AppleSEPXART::_handle_sep_driven_msg(AppleSEPXART::XARTMessage *): Replied to xART app:
0x27ab010000001513
** mounting realfs /dev/disk0s1s1 **
handle_mount:654: disk0s1s1 vol-uuid: 61706673-7575-6964-0001-766F6C756E00 block size: 4096 b
lock count: 62499909 (unencrypted; flags: 0x1; features: 20.0.12)
handle_mount:667: disk0s1s1 setting dev block size to 4096 from 512
sx_volume_group_update:7713: disk0s1s1 Volume System is not in a volume group
apfs_vfsop_mount:2354: mounted volume: System
** mounting fakefs /dev/disk0s1s0 **
set_cloneinfo_id_epoch:23515: disk0s1s0 set cloneinfo_id_epoch to 16
handle_mount:654: disk0s1s0 vol-uuid: 61706673-7575-6964-0004-766F6C756D07 block size: 4096 b
lock count: 62499909 (unencrypted; flags: 0x1; features: 8.0.2)
handle_mount:667: disk0s1s0 setting dev block size to 4096 from 512
sx_volume_group_update:7707: disk0s1s0 Volume System role 4 Not a System or data volume
apfs_vfsop_mount:2354: mounted volume: System
** copying files to fakefs (may take up to 10 minutes) **
CHG: _checkConnection() 'DIAP' data=1 connStatus=1 success=0
handlePowerSourceDetect: DIID=0, DIIG=0, DIAR=0, (0)
_debounceTimerCallback: timer callback now=699211600 hasVbus=1 port 1
Port-Lightning::handleIOBusCommand: Set InflowDisable for Power Handshake
handlePowerSourceDetect: DIID=1, DIIG=0, DIAR=0, (0)
IOAccessoryUSBConnect::handleAppleUSBCableDetect 1
IOAccessoryUSBConnect::handleUSB Power (VBUS) Present: 1, Physical Connection (CON_DET): 1
handleIOBusState: already in progress
AppleCD3L3612::setACCsConfig: (acc1Config=0x69646273, acc2Config=0x0)
AppleCD3L3612::setACCsConfig: orientationCheck=0, ACC_CTRL=0
handlePowerSourceDetect: DIID=1, DIIG=0, DIAR=0, (0)
Port-Lightning::getVBUSDetect: _vbusVoltageFunction returned device errorhandlePowerSourceDet
ect: DIID=1, DIIG=0, DIAR=0, (0)
Port-Lightning::getVBUSDetect: _vbusVoltageFunction returned device errorPort-Lightning::setP
owerHandshakeState: Clear InflowDisable, Power Handshake completed
handlePowerSourceDetect: DIID=0, DIIG=0, DIAR=0, (0)
handlePowerSourceDetect: DIID=0, DIIG=0, DIAR=0, (0)
IOAccessoryUSBC::_setPropertiesGated(): Setting USB Restricted Mode: YES... (m_mgrPrimaryPortI
D:1)
CHG: _checkConnection() 'DIAP' data=1 connStatus=1 success=0
CHG: _checkConnection() 'DIAP' data=1 connStatus=1 success=0
/Library/Caches/com.apple.xbs/Sources/AppleSMC/AppleSMC-633/AppleSMCEmbeddedCharger/AppleSMCC
harger.cpp:648 forcePowerStatus() Vbus=1 hasVBUS=1 _debounceUntil(=707770635 port 1
handlePowerSourceDetect: DIID=0, DIIG=0, DIAR=0, (0)
SMCNotifyExternalPowerChange: notify=0:0 VBUS1=1 dockNum 1 result (0)

直到看到最后的log：`rebooting in 5 seconds`


** mounting fakefs /dev/disk0s1s0 **
set_cloneinfo_id_epoch:25515: disk0s1s0 set cloneinfo_id_epoch to 16
handle_mount:654: disk0s1s0 vol-uuid: 61706673-7575-6964-0004-766F6C756D07 block size: 4096 b
lock count: 62499989 (unencrypted) flags: 0x1) features: 0.0.2)
handle_mount:667: disk0s1s0 setting dev block size to 4096 from 512
nx_volume_group_update:7707: disk0s1s0 Volume System role 4 Not a System or data volume
apfs_vfsop_mount:2354: mounted volume: System
** copying files to fakefs (may take up to 10 minutes) **
CHG: _checkConnection() 'D1AP' data=1 connStatus=1 success=0
handlePowerSourceDetect: D1ID=0, D1IG=0, D1AR=0, (0)
_debounceTimerCallback: timer callback now=699211680 hasVbus=1 port 1
Port-Lightning::handleIDBusCommand: Set InflowDisable for Power Handshake
handlePowerSourceDetect: D1ID=1, D1IG=0, D1AR=0, (0)
IOAccessoryUSBConnectShim: AppleUSBCableDetect 1
IOAccessoryUSBConnectShim: USB Power (VBUS) Present: 1, Physical Connection (CON_DET): 1
handleIDBusState: already in progress
AppleCD3L1612::setACCxConfig: (accConfig=0x69646273, acc2Config=0x0)
AppleCD3L1612::setACCxConfig: orientationCheck=0, ACC_CTRL=0
handlePowerSourceDetect: D1ID=1, D1IG=0, D1AR=0, (0)
Port-Lightning::getVBUSDetect: _vbusVoltageFunction returned device errorhandlePowerSourceDet
ect: D1ID=1, D1IG=0, D1AR=0, (0)
Port-Lightning::getVBUSDetect: _vbusVoltageFunction returned device errorPort-Lightning::setP
owerHandshakeState: Clear InflowDisable, Power Handshake completed
handlePowerSourceDetect: D1ID=0, D1IG=0, D1AR=0, (0)
handlePowerSourceDetect: D1ID=0, D1IG=0, D1AR=0, (0)
IOAccessoryTAM::_setPropertiesGated(): Setting USB Restricted Mode: YES... (m_mgrPrimaryPortI
D: 1)
CHG: _checkConnection() 'D1AP' data=1 connStatus=1 success=0
CHG: _checkConnection() 'D1AP' data=1 connStatus=1 success=0
/Library/Caches/com.apple.xbs/Sources/AppleSMC/AppleSMC-633/AppleSMCEmbeddedCharger/AppleSMCC
harger.cpp:645 forcePowerStatus() vbus=1 _hasVBUS=1 _debounceUntil=707778635 port 1
handlePowerSourceDetect: D1ID=0, D1IG=0, D1AR=0, (0)
AppleSMCNotifyExternalPowerChange: notify=0:0 VBUS1=1 dockNum 1 result (0)
[AWS2:client1] New BlogBuf low 7
[AWS2:client1] New BlogBuf low 6
virtual bool CoreAnalyticsPipe::checkForWork():174:starting CoreAnalyticsMessenger
void CoreAnalyticsHub::handleNagTimerExpiry(IOTimerEventSource *):627:messageClients of 3 av
ailable events
[SMC:charger.cpp:1954] Not charging:0
tx_flush:1171: disk0s1 xid 5555239 tx stats: # 20 finish 2 enter 79955 wait 17 5299us close 0
us flush 14113us
000144.304837 wlan0.AE[1] handleAdjustBusy01537:AdjustBusy timeout in 120000 ms! busystate 0
** syncing filesystems **
tx_flush:1171: disk0s1 xid 5555259 tx stats: # 40 finish 3 enter 79134 wait 35 5759us close 0
us flush 15375us
** unmounting filesystems **
apfs_vfsop_unmount:2565: disk0s1s1 disk0s1: unmounting volume
apfs_vfsop_unmount:2598: disk0s1s1 waiting for purgatory cleaner to finish
apfs_vfsop_unmount:2932: disk0s1 nx_num_vols_mounted is 8
apfs: total snm allocated: 45271927 (43 mb);
apfs_vfsop_unmount:2945: all done. going home. (numMountedAPFSVolumes 8)
apfs_vfsop_unmount:2565: disk0s1s0 disk0s1: unmounting volume
apfs_vfsop_unmount:2598: disk0s1s0 waiting for purgatory cleaner to finish
nx_volume_group_update:7707: disk0s1s0 Volume System role 4 Not a System or data volume
apfs_vfsop_unmount:2932: disk0s1 nx_num_vols_mounted is 7
apfs: total snm allocated: 44298378 (42 mb);
apfs_vfsop_unmount:2945: all done. going home. (numMountedAPFSVolumes 7)
apfs_stop_bg_work:1065: disk0s1s5 Volume Hardware is unmounting, stop any bg work
unmount(/private/var/hardware): Resource busy -- try 'diskutil unmount'
apfs_stop_bg_work:1065: disk0s1s7 Volume Update is unmounting, stop any bg work
apfs_vfsop_unmount:2565: disk0s1s7 disk0s1: unmounting volume
apfs_vfsop_unmount:2598: disk0s1s7 waiting for purgatory cleaner to finish
nx_volume_group_update:7707: disk0s1s7 Volume Update role c0 Not a System or data volume
apfs_vfsop_unmount:2932: disk0s1 nx_num_vols_mounted is 6
apfs: total snm allocated: 44289878 (42 mb);
apfs_vfsop_unmount:2945: all done. going home. (numMountedAPFSVolumes 6)
apfs_stop_bg_work:1065: disk0s1s4 Volume Baseband Data is unmounting, stop any bg work
apfs_vfsop_unmount:2565: disk0s1s4 disk0s1: unmounting volume
apfs_vfsop_unmount:2598: disk0s1s4 waiting for purgatory cleaner to finish
nx_volume_group_update:7707: disk0s1s4 Volume Baseband Data role 40 Not a System or data volu
me
apfs_vfsop_unmount:2932: disk0s1 nx_num_vols_mounted is 5
apfs: total snm allocated: 44287503 (42 mb);
apfs_vfsop_unmount:2945: all done. going home. (numMountedAPFSVolumes 5)
apfs_stop_bg_work:1065: disk0s1s2 Volume Data is unmounting, stop any bg work
unmount(/private/var): Resource busy -- try 'diskutil unmount'
unmount(/private/preboot): Resource busy -- try 'diskutil unmount'
apfs_stop_bg_work:1065: disk0s1s3 Volume xART is unmounting, stop any bg work
unmount(/private/xarts): Resource busy -- try 'diskutil unmount'
** Done setting up fakefs **
waitpid for '/cores/binpack/bin/sh /cores/create_fakefs.sh /dev/disk0s1s0 0 ' returned: 2
Rebooting in 5 seconds

iPhone会继续重启，然后进入桌面

此时iPhone桌面中，还没有安装palera1n的app。

第三步：palera1n -f，首次会安装palera1n的app

继续去：

```
palera1n -f
```

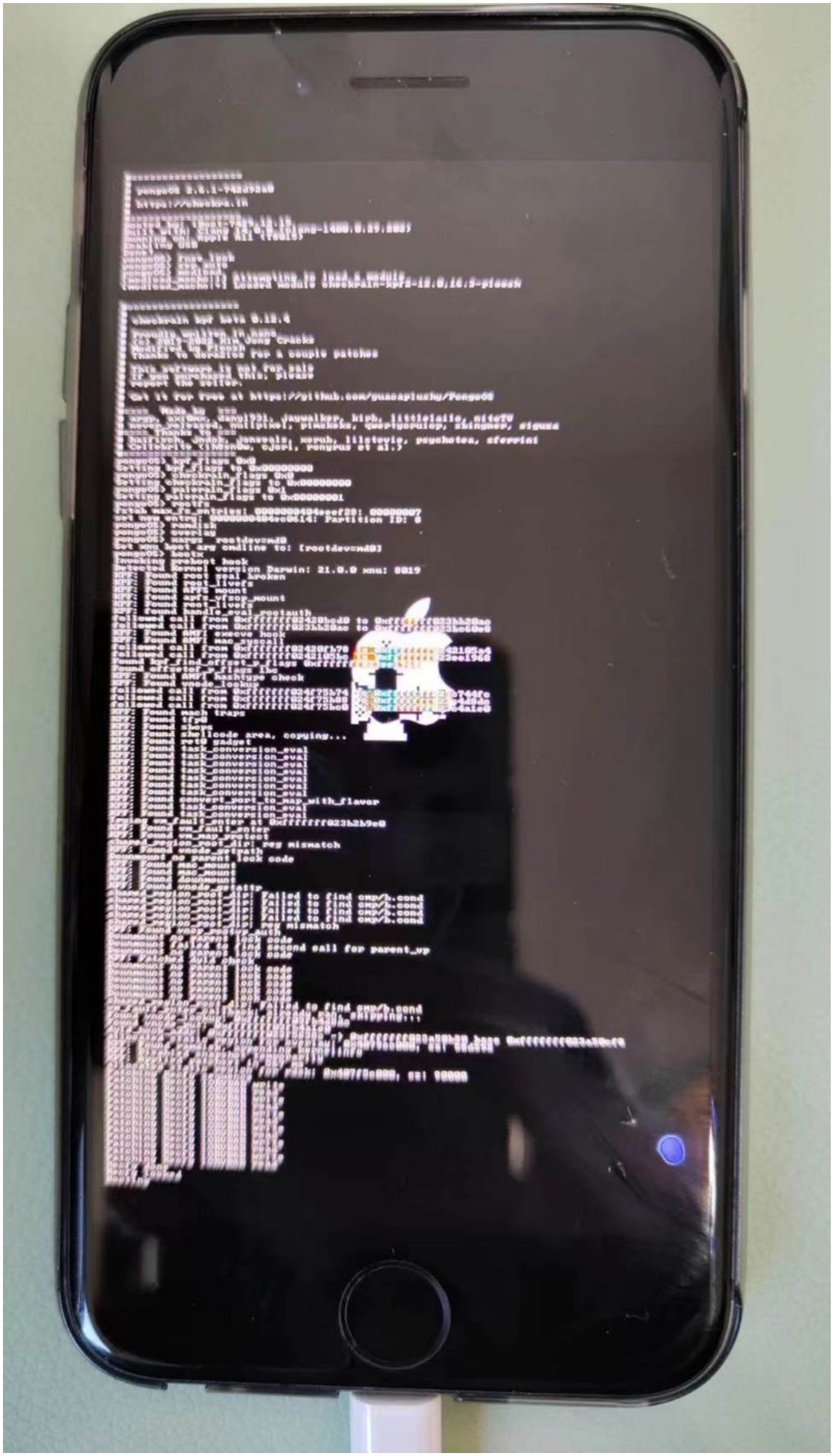
去：启动设备

继续按照提示，操作iPhone进入DFU模式

详细log日志：

```
crifan@licrifandeMacBook-Pro ~/dev/dev_tool/reverse_security/iOS/palera1n palera1n -f
# == palera1n-c ==
#
# Made by: Nick Chan, Ploosh, Mineek, Nebula, llsc12
#
# Thanks to: dora2ios, pythonplayer, tihmstar, nikias
# (libimobiledevice), checkra1n team (Siguza, axi0mx, littlelailo
# et al.), Procursus Team (Hayden Seay, Cameron Katri, Keto et.al)

 - [02/27/23 14:48:09] Info : Waiting for devices
 - [02/27/23 14:48:09] Info : Telling device with udid abdc0dd961c3cb96f5c4afe109de4eb48b88433a to enter recovery mo
de immediately
 - [02/27/23 14:48:20] Info : Press Enter when ready for DFU mode

Get ready (0)
Hold volume down + side button (0)
Hold volume down button (5)
 - [02/27/23 14:49:55] Info : Device entered DFU mode successfully
 - [02/27/23 14:49:56] Info : About to execute checkra1n
#
# Checkra1n 0.1337.1
#
# Proudly written in nano
# (c) 2019-2023 Kim Jong Cracks
#
#========  Made by  =======
# argp, axi0mx, danyl931, jaywalker, kirb, littlelailo, nitoTV
# never_released, nullpixel, pimskeks, qwertyoruiop, sbingner, siguza
#======== Thanks to =======
# haifisch, jndok, jonseals, xerub, lilstevie, psychotea, sferrini
# Cellebrite (ih8sn0w, cjori, ronyrus et al.)
#============================

 - [02/27/23 14:49:56] Verbose : Starting thread for Apple TV 4K Advanced board
 - [02/27/23 14:49:56] Info : Waiting for DFU mode devices
 - [02/27/23 14:49:56] Verbose : DFU mode device found
 - [02/27/23 14:49:56] Info : Checking if device is ready
 - [02/27/23 14:49:56] Verbose : Attempting to perform checkm8 on 8015 11
 - [02/27/23 14:49:56] Info : Setting up the exploit
 - [02/27/23 14:49:56] Verbose :    checkm8 setup stage
 - [02/27/23 14:49:56] Verbose : Entered initial checkm8 state after 1 steps
 - [02/27/23 14:49:56] Verbose : Stalled input endpoint after 6 steps
 - [02/27/23 14:49:56] Verbose : DFU mode device disconnected
 - [02/27/23 14:49:56] Verbose : DFU mode device found
 - [02/27/23 14:49:56] Verbose :    checkm8 trigger stage
 - [02/27/23 14:49:57] Info : Checkmate
 - [02/27/23 14:49:57] Verbose : Device should now reconnect in download mode
 - [02/27/23 14:49:57] Verbose : DFU mode device disconnected
 - [02/27/23 14:50:04] Info : Entered download mode
 - [02/27/23 14:50:04] Verbose : Download mode device found
 - [02/27/23 14:50:04] Info : Booting PongoOS...
 - [02/27/23 14:50:06] Info : Found PongoOS USB Device
 - [02/27/23 14:50:06] Info : Booting Kernel...
```

iPhone中启动输出日志，其中屏幕中间可见 苹果的logo图标（其中嵌入了一个checkra1n的灯塔图标？）：



期间会自动安装：palera1n的app

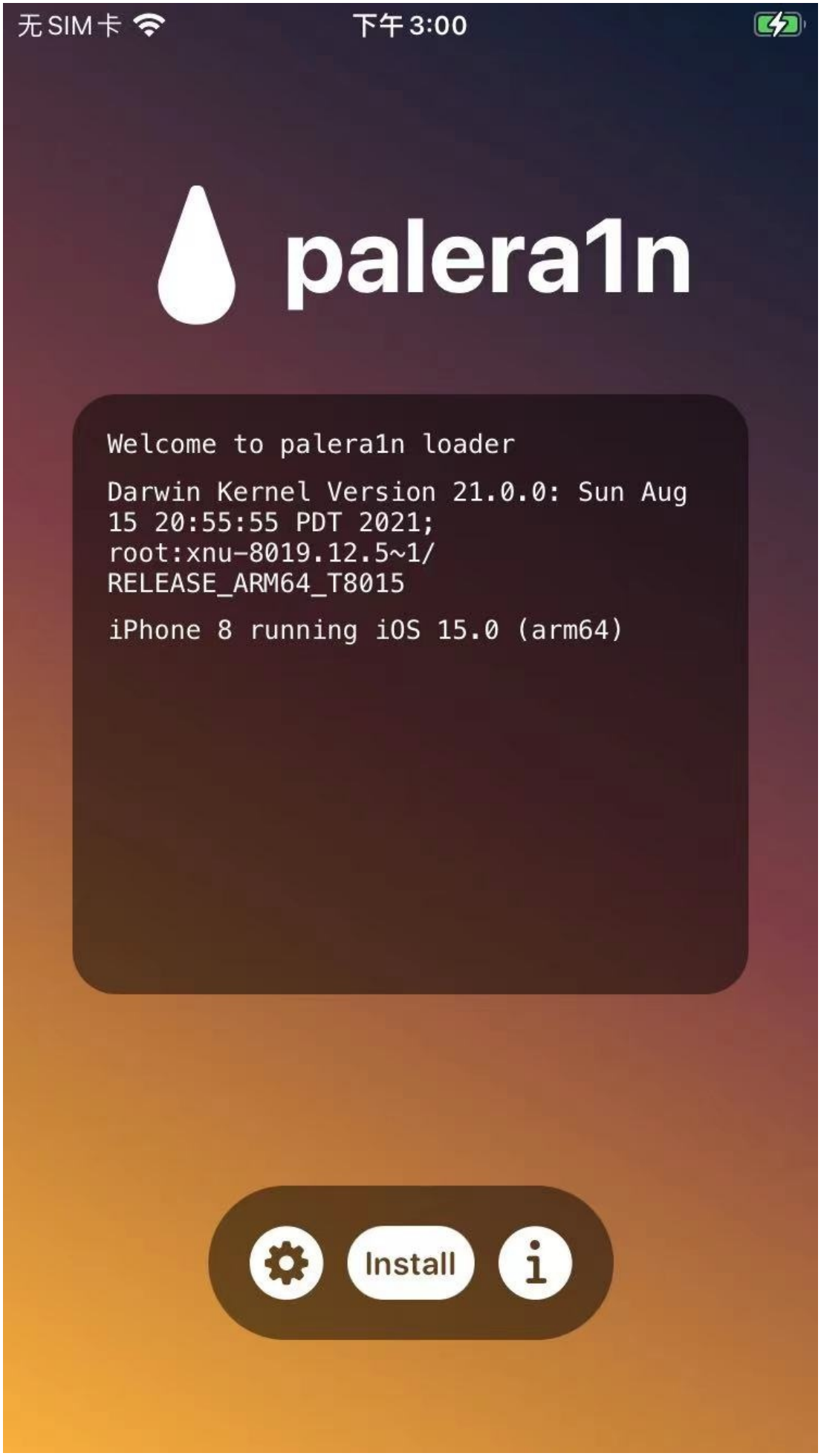
进入iPhone桌面后，可以看到：

`palera1n的app` = `palera1n loader` = paleran的图标


无SIM卡
下午2:50
天气
查找
快捷指令
家庭
通讯录
文件
股市
翻译
图书
iTunes Store
Watch
提示
实用工具
爱思极速版
微信
palera1n

第四步：进入palera1n的app去Install安装

打开palera1n的app后，进入主页，能看到有个Install按钮


无SIM卡
下午3:00
palera1n
Welcome to palera1n loader
Darwin Kernel Version 21.0.0: Sun Aug 15 20:55:55 PDT 2021; root:xnu-8019.12.5~1/RELEASE_ARM64_T8015
iPhone 8 running iOS 15.0 (arm64)
Install

此处会显示：

- 当前iPhone信息
  - iPhone 8 running i0s 15.0 (arm64)

点击Install，会继续越狱过程，输出log过程，直到最后：

- `Finished installing! Enjoy!`

期间会下载和安装：

- bootstrap.tar
- sileo.deb
- straprepo.deb

对应地址分别是：

- 此处的普通越狱
  - https://cdn.nickchan.lol/palera1n/loader/assets/bootstrap.tar
  - https://cdn.nickchan.lol/palera1n/loader/assets/sileo.deb
  - https://cdn.nickchan.lol/palera1n/loader/assets/straprepo.deb
- 如果是rootless越狱
  - https://cdn.nickchan.lol/palera1n/loader/assets/rootless/bootstrap.tar
  - https://cdn.nickchan.lol/palera1n/loader/assets/rootless/palera1nrepo.deb
  - https://cdn.nickchan.lol/palera1n/loader/assets/rootless/sileo.deb

点击：`Respring`=重启桌面

然后桌面上即可看到：`Sileo`了


无SIM卡
下午3:39
天气
查找
快捷指令
家庭
通讯录
文件
股市
翻译
图书
iTunes Store
Watch
提示
实用工具
爱思极速版
信息
音乐
palera1n
Sileo

至此，palera1n越狱过程就结束了。

可以愉快的用Sileo去安装各种越狱插件了。

常见问题

Warning Whoops device did not enter DFU mode

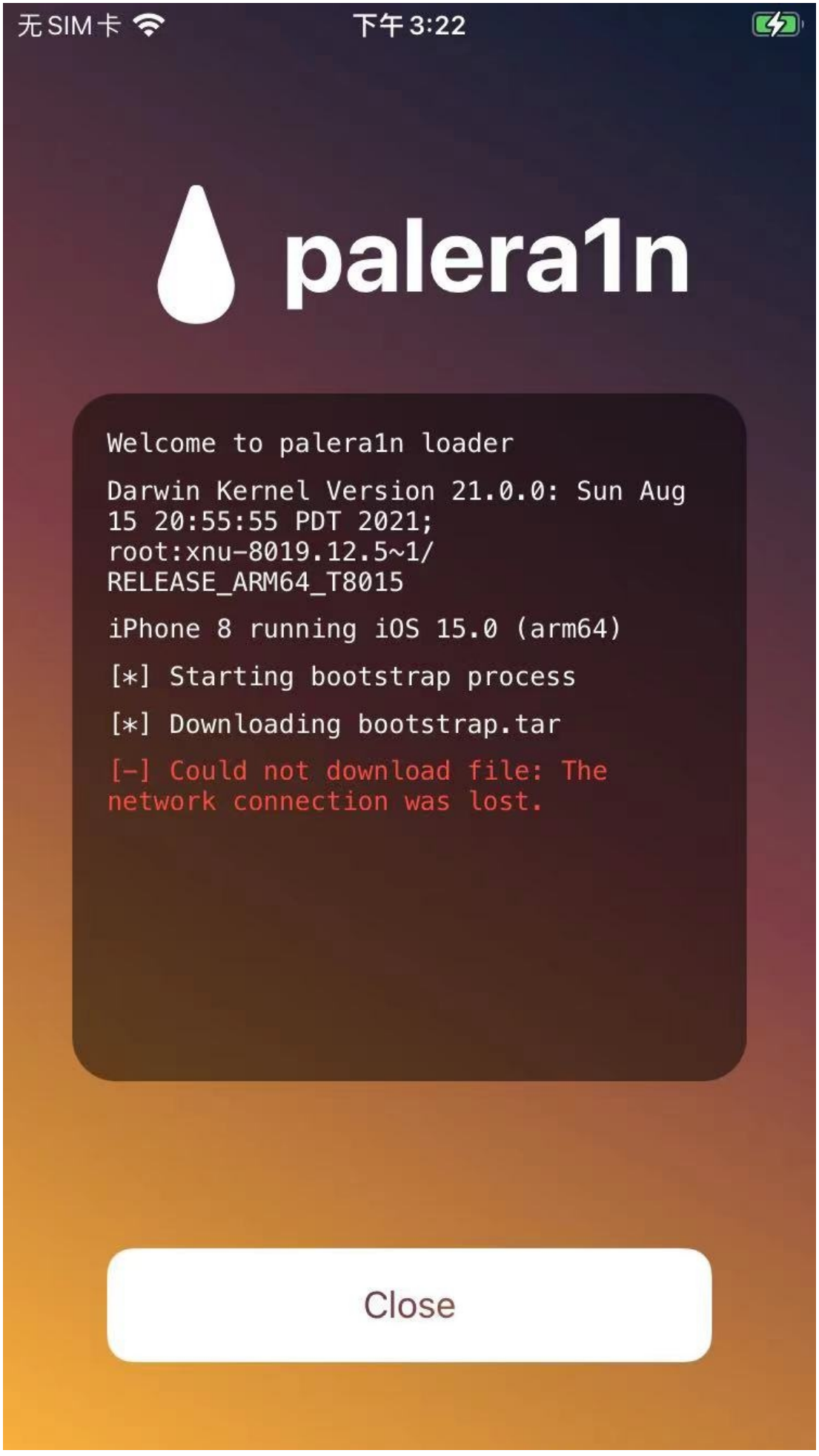
现象：尝试进入DFU模式时遇到：`<Warning>: Whoops, device did not enter DFU mode`

原因：在操作`Hold volume down + side button`和`Hold volume down button`期间，`volume down`键，被我松掉了

解决办法：期间按住`volume down`不要松手，持续按住，即可顺利进入DFU模式

Could not download file The network connection was lost

问题：


无SIM卡
下午3:22
palera1n
Welcome to palera1n loader
Darwin Kernel Version 21.0.0: Sun Aug 15 20:55:55 PDT 2021; root:xnu-8019.12.5~1/RELEASE_ARM64_T8015
iPhone 8 running iOS 15.0 (arm64)
[*] Starting bootstrap process
[*] Downloading bootstrap.tar
[-] Could not download file: The network connection was lost.
Close

- 解决办法
  - 多试几次

palra1n的help语法

```
crifan@licrifandeMacBook-Pro ~/dev/dev_tool/reverse_security/iOS/palera1n palera1n --help
Usage: palera1n [-cCdDEfhlLnOpRsvV] [-e boot arguments] [-k Pongo image] [-o overlay file] [-r ramdisk file] [-K KPF
file] [-i checkra1n file]
Copyright (C) 2023, palera1n team, All Rights Reserved.

iOS/iPadOS 15+ arm64 jailbreaking tool

    --version               Print version
    --force-revert              Remove jailbreak
    -B, --setup-partial-fakefs      Setup partial fakefs
    -c, --setup-fakefs          Setup fakefs
    -d, --demote              Demote
    -D, --dfuhelper           Exit after entering DFU
    -e, --boot-args boot arguments    XNU boot arguments
    -E, --enter-recovery          Enter recovery mode
    -f, --fakefs              Boots fakefs
    -h, --help              Show this help
    -i, --override-checkra1n file       Override checkra1n
    -k, --override-pongo file       Override Pongo image
    -K, --override-kpf file       Override kernel patchfinder
    -l, --rootless              Boots rootless. This is the default
    -L, --jbinit-log-to-file      Make jbinit log to /cores/jbinit.log (can be read from sandbox while jailbroken)
    -n, --exit-recovery         Exit recovery mode
    -I, --device-info           Print info about the connected device
    -o, --override-overlay file       Override overlay
    -O, --disable-ohio          Disable Ohio
    -p, --pongo-shell           Boots to PongoOS shell
    -P, --pongo-full          Boots to a PongoOS shell with default images already uploaded
    -r, --override-ramdisk file       Override ramdisk
    -R, --reboot-device         Reboot connected device in normal mode
    -s, --safe-mode             Enter safe mode
    -v, --debug-logging         Enable debug logging
        This option can be repeated for extra verbosity.
    -V, --verbose-boot          Verbose boot

Environmental variables:
    TMPDIR        temporary diretory (path the built-in checkra1n will be extracted to)
```

palera1n -I

用palera1n查看当前连接的iPhone的信息：

```
crifan@licrifandeMacBook-Pro ~/dev/dev_tool/reverse_security/iOS/palera1n palera1n -I
Mode: normal
ProductType: iPhone10,1
Architecture: arm64
Version: 15.0
DisplayName: iPhone 8 (Global)
```

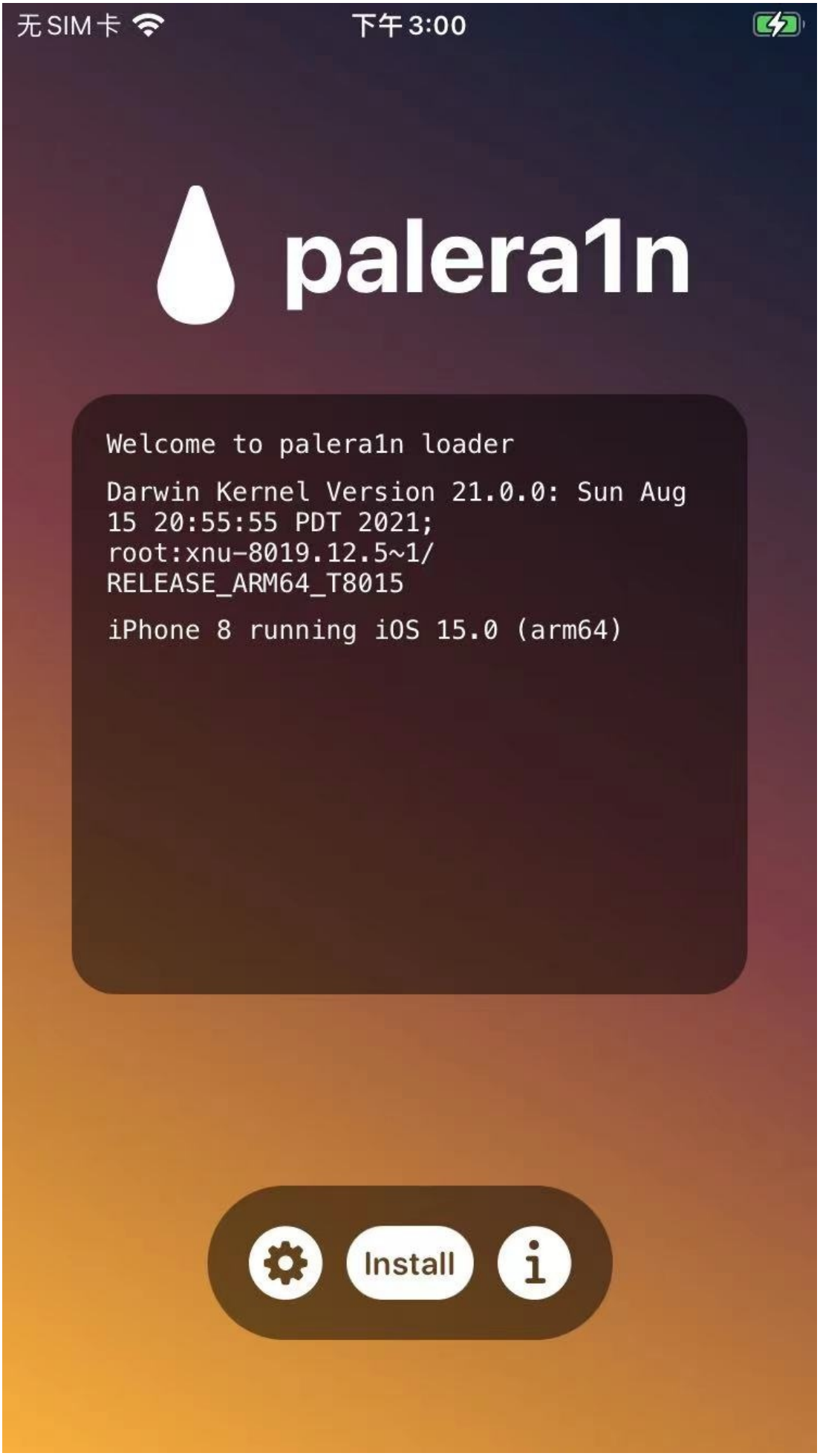
palera1n loader app

- palera1n的app = palera1n loader = palera1n loader app

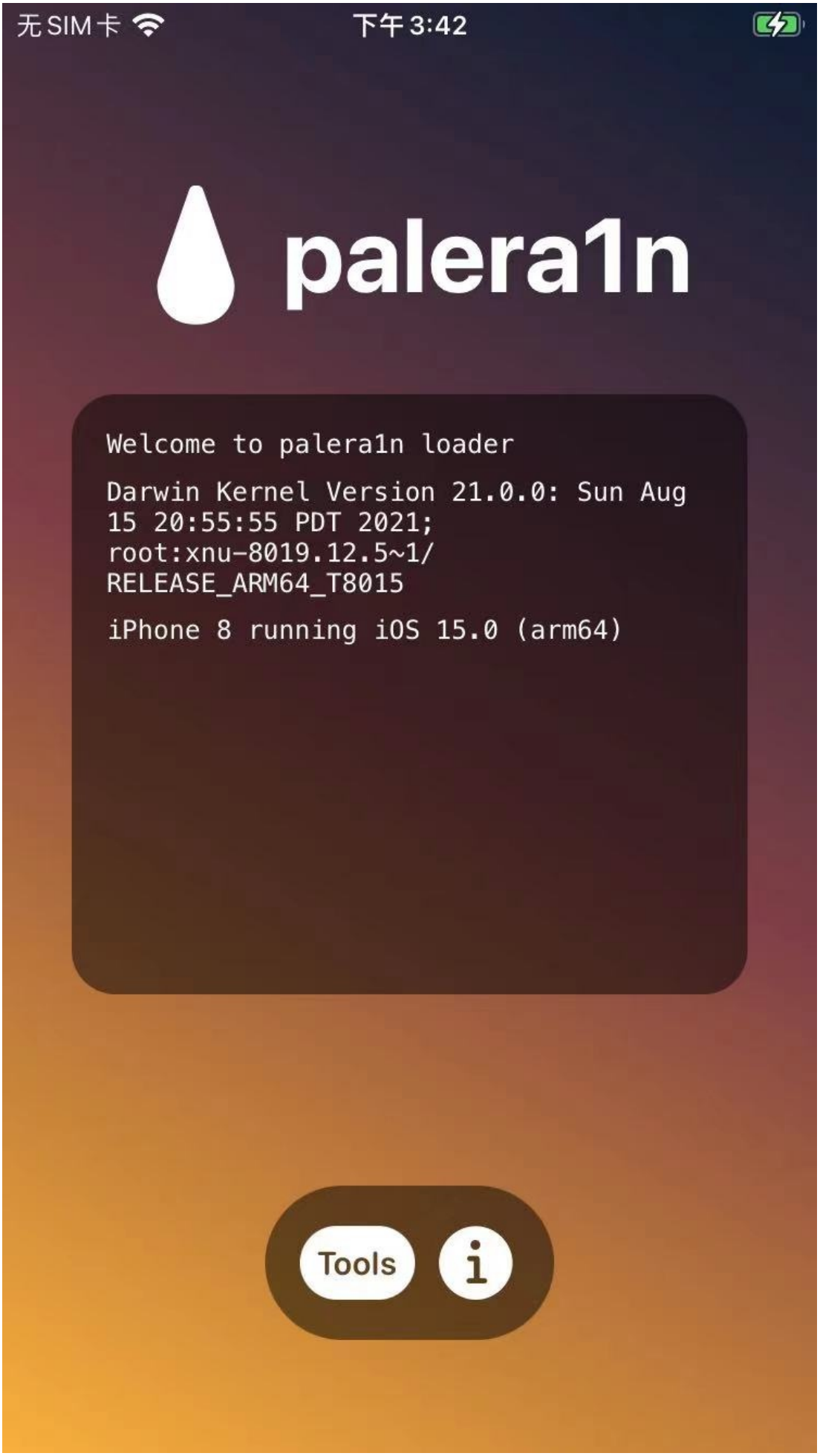
打开palera1n的app后：

首页

- 如果中间出现Install按钮：说明越狱过程还没结束
  -


无SIM卡
下午3:00
palera1n
Welcome to palera1n loader
Darwin Kernel Version 21.0.0: Sun Aug 15 20:55:55 PDT 2021; root:xnu-8019.12.5~1/RELEASE_ARM64_T8015
iPhone 8 running iOS 15.0 (arm64)
Install

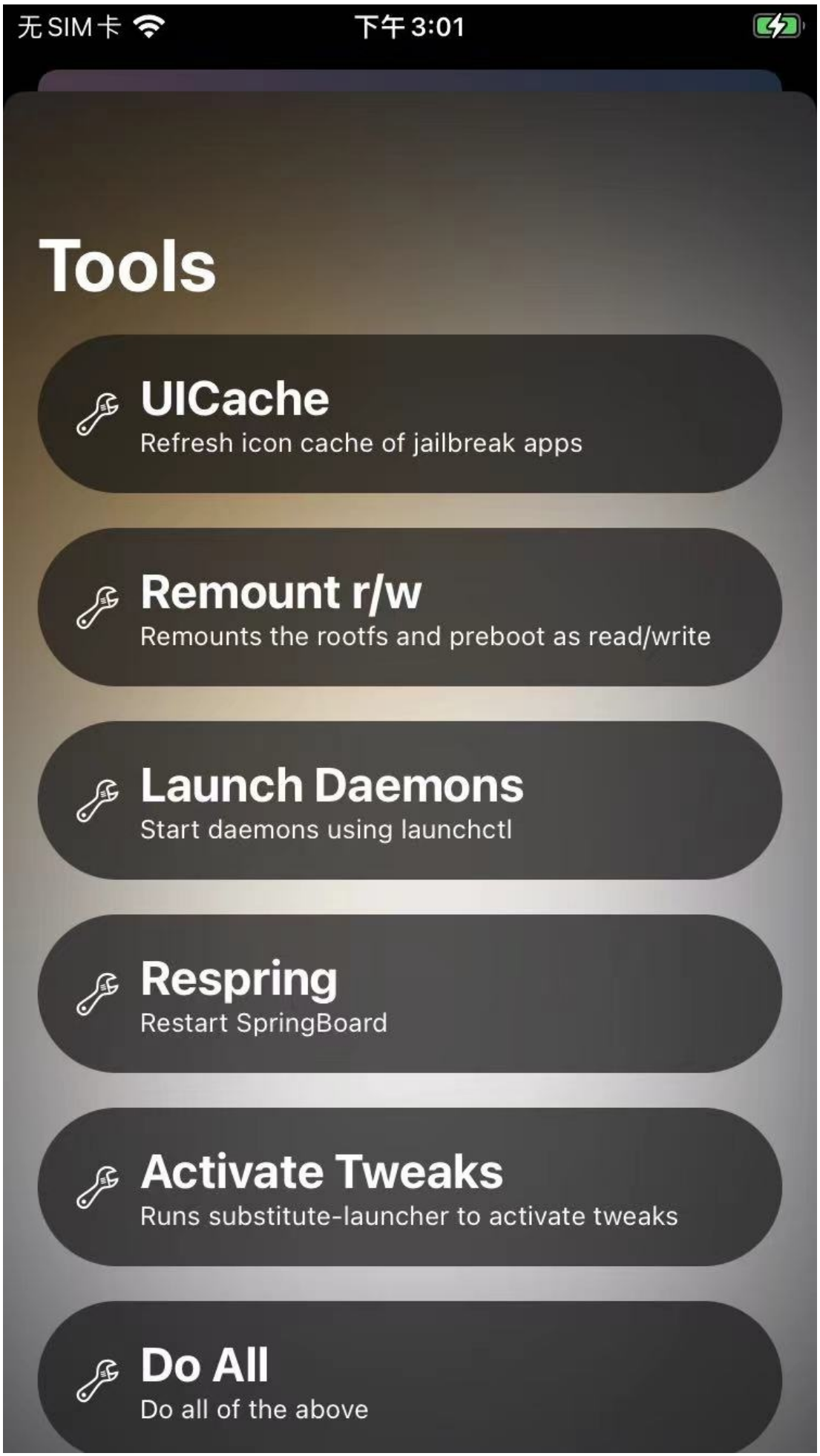
- 如果中间没有了Install按钮：说明已经越狱成功
  -

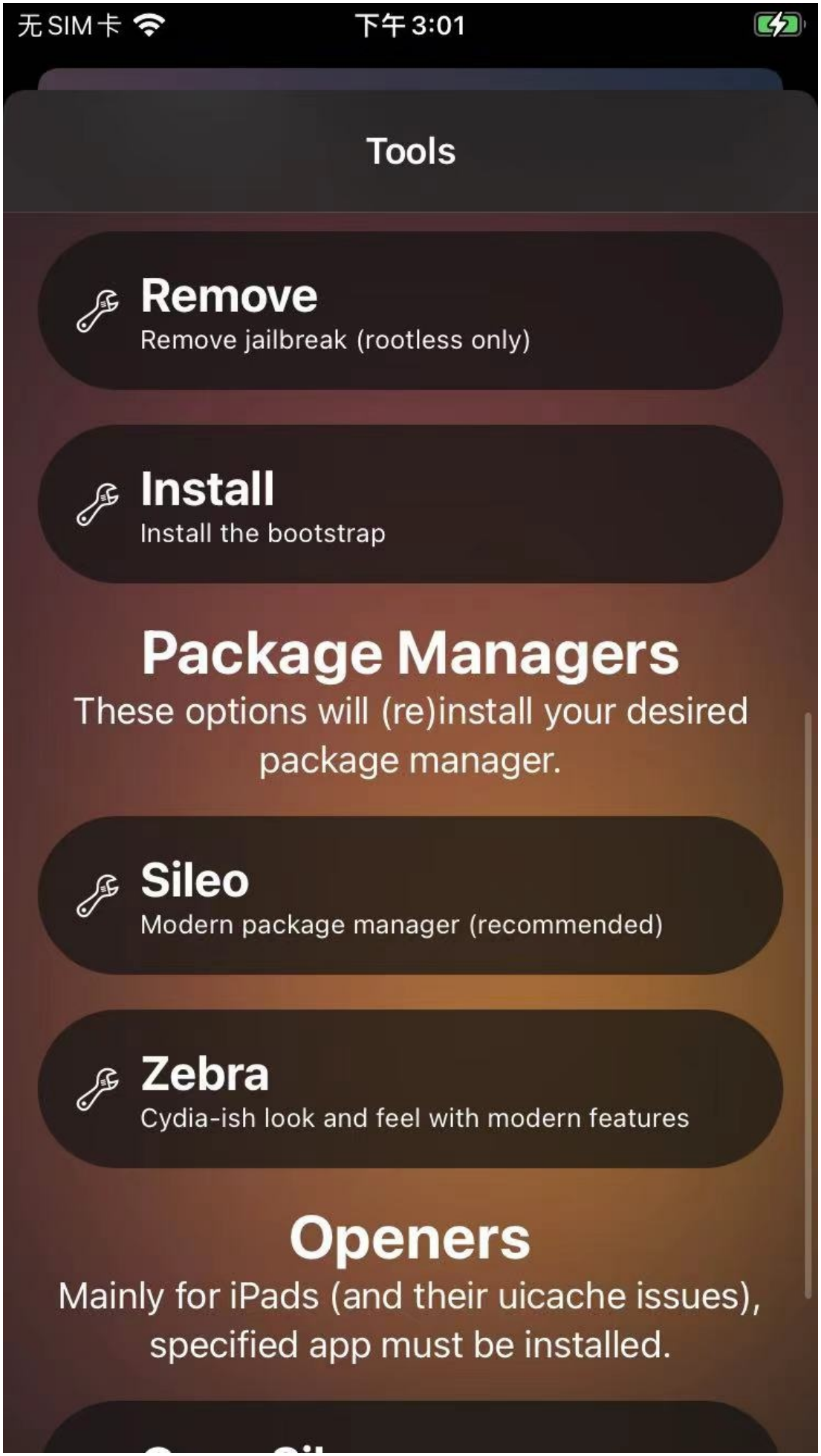

无SIM卡
下午3:42
palera1n
Welcome to palera1n loader
Darwin Kernel Version 21.0.0: Sun Aug 15 20:55:55 PDT 2021; root:xnu-8019.12.5~1/RELEASE_ARM64_T8015
iPhone 8 running iOS 15.0 (arm64)
Tools

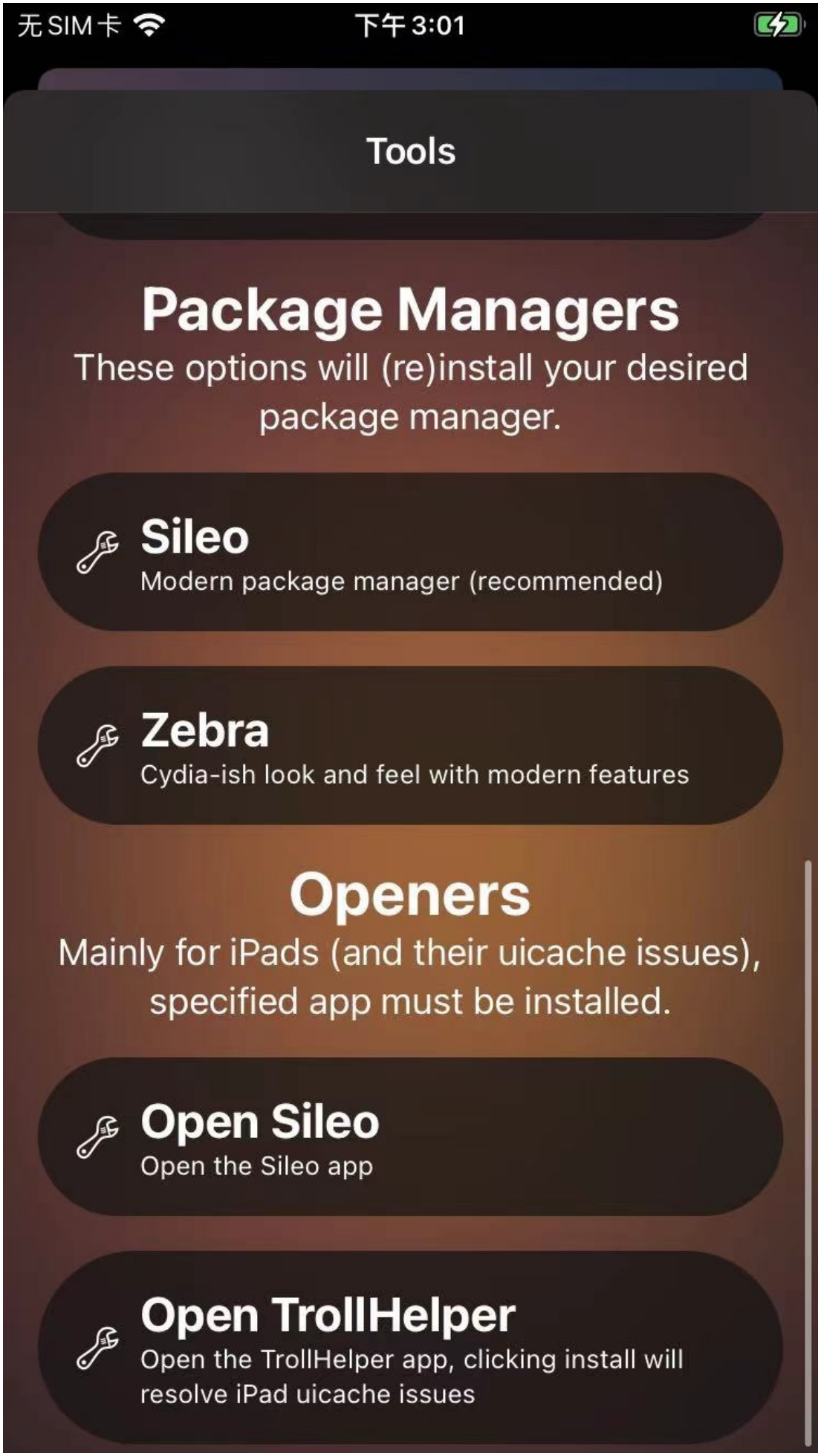
去看看palera1n的app的其他信息：

Tools

点击左边的齿轮图标=设置=工具=Tools，会打开Tools页面：

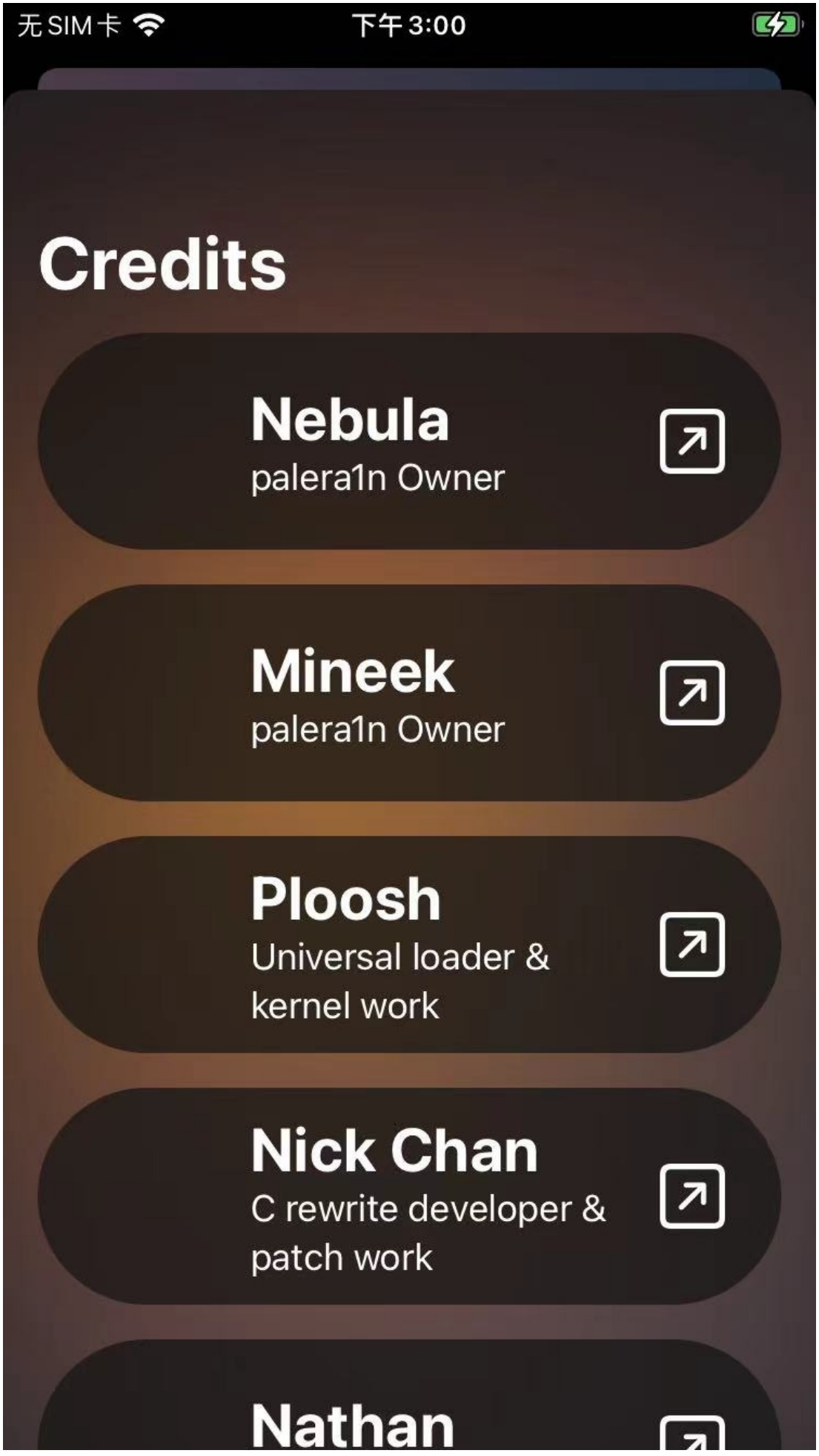

无SIM卡
下午3:01
Tools
UICache
Refresh icon cache of jailbreak apps
Remount r/w
Remounts the rootfs and preboot as read/write
Launch Daemons
Start daemons using launchctl
Respring
Restart SpringBoard
Activate Tweaks
Runs substitute-launcher to activate tweaks
Do All
Do all of the above

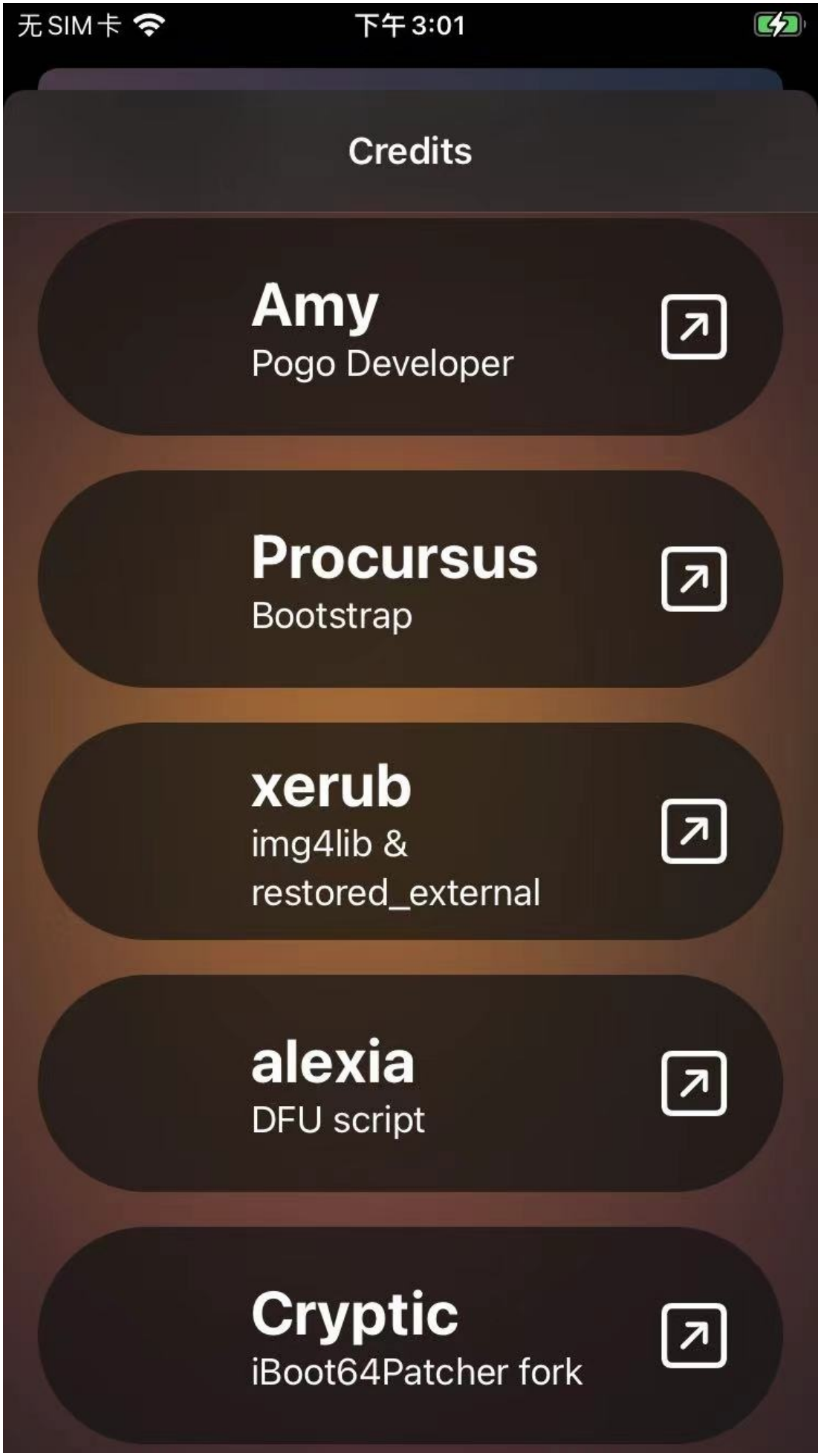

无SIM卡
下午3:01
Tools
Remove
Remove jailbreak (rootless only)
Install
Install the bootstrap
Package Managers
These options will (re)install your desired package manager.
Sileo
Modern package manager (recommended)
Zebra
Cydia-ish look and feel with modern features
Openers
Mainly for iPads (and their uicache issues), specified app must be installed.


无SIM卡
下午3:01
Tools
Package Managers
These options will (re)install your desired package manager.
Sileo
Modern package manager (recommended)
Zebra
Cydia-ish look and feel with modern features
Openers
Mainly for iPads (and their uicache issues), specified app must be installed.
Open Sileo
Open the Sileo app
Open TrollHelper
Open the TrollHelper app, clicking install will resolve iPad uicache issues

Credits

点击右边的i=info，会打开Credits页面：


无SIM卡
下午3:00
Credits
Nebula
palera1n Owner
Mineek
palera1n Owner
Ploosh
Universal loader &
kernel work
Nick Chan
C rewrite developer &
patch work
Nathan


无SIM卡
下午3:01
Credits
Amy
Pogo Developer
Procursus
Bootstrap
xerub
img4lib &
restored_external
alexia
DFU script
Cryptic
iBoot64Patcher fork

palera1n越狱后

能看到palera1n的进程

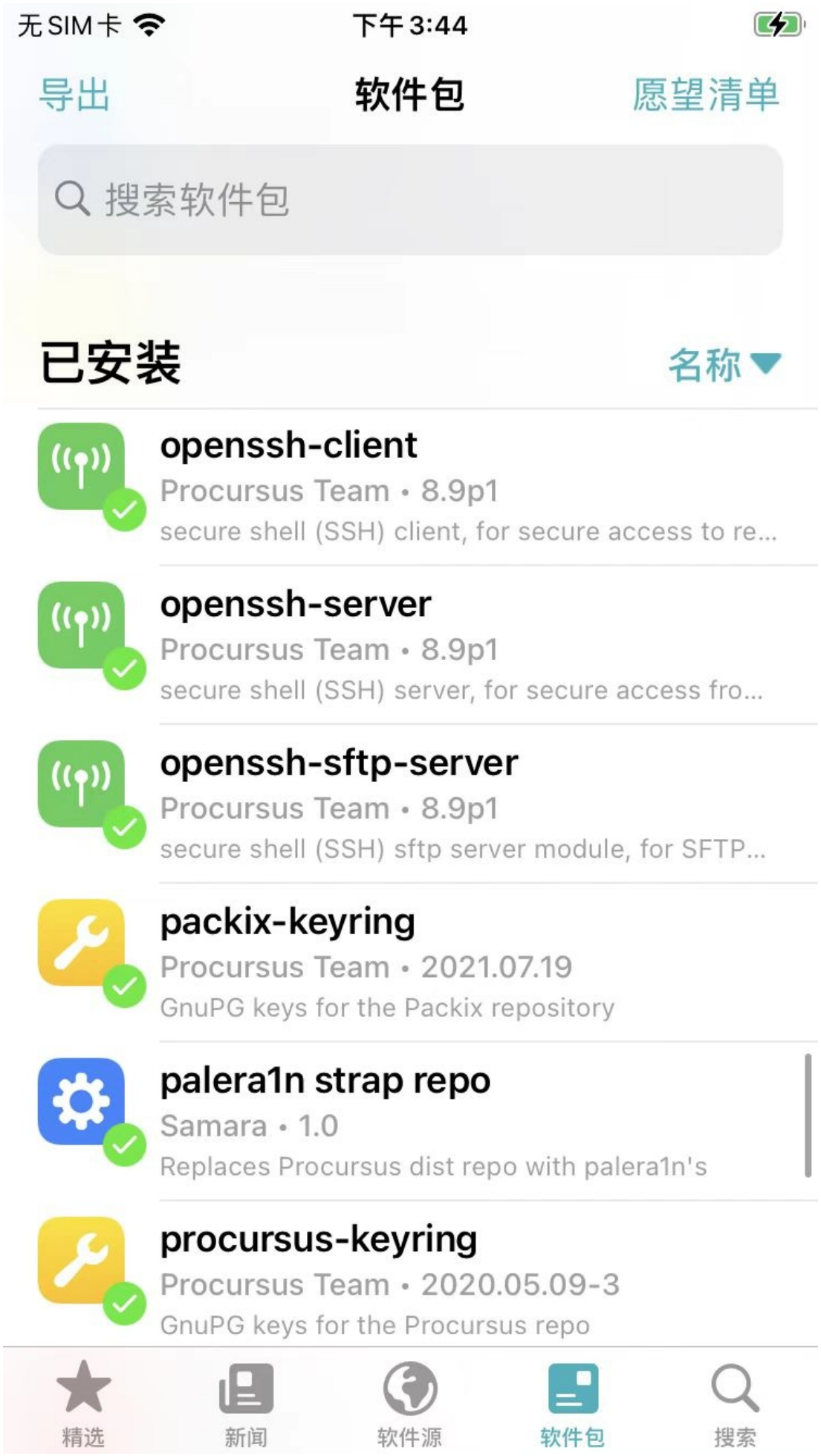
Mac中通过frida-ps查看的：

```
crifan@licrifandeMacBook-Pro ~/dev/dev_tool/reverse_security/iOS/Filza/dep_tools frida-ps -U
PID  Name
----  ------------------------------------------------------------
''
1660
      Sileo
2113
      palera1n
2118
      信息
''
```

ssh可以直接使用

因为已自动安装了对应OpenSSH的相关插件：

- openssh-client
- openssh-server
- openssh-sftp-server


无SIM卡
下午3:44
导出
软件包
愿望清单
搜索软件包
已安装
名称
openssh-client
Procursus Team • 8.9p1
secure shell (SSH) client, for secure access to re...
openssh-server
Procursus Team • 8.9p1
secure shell (SSH) server, for secure access fro...
openssh-sftp-server
Procursus Team • 8.9p1
secure shell (SSH) sftp server module, for SFTP...
packix-keyring
Procursus Team • 2021.07.19
GnuPG keys for the Packix repository
palera1n strap repo
Samara • 1.0
Replaces Procursus dist repo with palera1n's
procursus-keyring
Procursus Team • 2020.05.09-3
GnuPG keys for the Procursus repo
精选
新闻
软件源
软件包
搜索

初始化ssh环境

第一次连接：

```
ssh root@192.168.2.13
```

- 说明
  - `192.168.2.13`：你的iPhone的IP
    - 和你的Mac使用同一个WiFi(网络)

输入：`yes`

再输入（OpenSSH的默认）密码：`alpine`

ssh免密登录

```
ssh-copy-id root@192.168.2.13
```

输入密码，即可：

-》之后每次直接连接：

```
ssh root@192.168.2.13
```

而无需密码。

TODO：加上palera1n的rootless越狱后，ssh无法直接使用的过程。

palera1n越狱后，重启丢失越狱，如何恢复越狱

palera1n越狱后的设备，（如果，由于各种原因，主动的或者被动的，不得不）重启后，会：丢失越狱，所以要：恢复越狱

下面是，palera1n恢复越狱的过程：

```
palera1n -f
```

继续按照提示操作即可，和第一次越狱的步骤类似。

- 核心步骤：
  - Enter回车键
  - 给iPhone进入DFU模式
    - Hold volume down + side button (0)
    - Hold volume down button (3)
  - 如何判断恢复越狱成功?
    - palera1n的图标正常 + 进入palera1n后没有 `Install` 按钮
      - 表示已成功恢复越狱

palera1n越狱后的常见问题

缺少常用解压缩工具

现象：palera1n越狱后，没有自动安装常用的解压缩工具(插件)，即zip unzip gzip unrar p7zip bzip2等等，需要自己去安装

解决办法：

有2种方式：

- 手动找到对应的deb地址，并下载得到deb文件，自己手动用dpkg安装
  - 去哪里找deb地址?
    - 去这里：
      - https://www.ios-repo-updates.com
    - 贴出部分deb地址
      - zip
        - http://tigisoftware.com/rootless/debs/zip_3.0_iphoneos-arm.deb
      - unzip
        - https://apt.procurs.us/pool/main/iphoneos-arm64/1700/unzip/unzip_6.0-27_iphoneos-arm.deb
      - gzip
        - https://apt.procurs.us/pool/main/iphoneos-arm64/1700/gzip/gzip_1.12_iphoneos-arm.deb
      - unrar
        - https://apt.procurs.us/pool/main/iphoneos-arm64/1700/unrar_6.1.4_iphoneos-arm.deb
      - p7zip
        - https://repo.chimera.sh/debs/p7zip_16.02_iphoneos-arm.deb
      - bzip2
        - https://apt.procurs.us/pool/main/iphoneos-arm64/1700/bzip2_1.0.8_iphoneos-arm.deb
  - 如何安装?
    - 去iPhone中安装
      - Mac通过ssh登录iPhone，或进入iPhone中的终端
    - 安装命令
      - dpkg -i xxx.deb
- Sileo中，找到对应的软件源，添加软件源，搜索对应插件，去安装
  - （包含这些解压缩工具的）软件源的地址：
    - https://apt.procurs.us

XinaA15

- XinaA15
  - 是什么：iPhone越狱工具
    - XinaA15 is a `semi-untethered` jailbreak made for `A12+` devices running iOS `15.0` up to `15.1.1`
  - 旧称
    - XinA12
    - XinA15
  - 作者
    - 中文名：`朱心浪`
    - 英文名：`xina520`
      - 联系方式
        - Twitter：https://twitter.com/xina520
  - 官网
    - https://xina.ss03.cn/
  - 官方文档
    - jacksight/xina520_official_jailbreak: This is the official download website for xina520's jailbreak
      - 该文档的 `优化版` （修复错别字等）
        - NotDarkn/XinaA15: XinaA15: A semi-untethered jailbreak for iOS 15.0-15.1.1
          - Compatibility · NotDarkn/XinaA15 Wiki
          - Warnings · NotDarkn/XinaA15 Wiki
          - XinaA15/-INSTALL.md at main · NotDarkn/XinaA15 · GitHub
          - XinaA15/-UPDATE.md at main · NotDarkn/XinaA15 · GitHub
          - XinaA15/-REMOVE.md at main · NotDarkn/XinaA15 · GitHub
          - Fixes · NotDarkn/XinaA15 Wiki
  - 支持设备
    - 机型：`iPhone` / `iPad`
    - CPU芯片：`A12` ~ `A15`
    - iOS版本：`iOS 15.0` ~ `iOS 15.1.1`
      - 具体包括
        - `iOS 15.1.1`
        - `iOS 15.1`
        - `iOS 15.0.3`
        - `iOS 15.0.2`
        - `iOS 15.0.1`
        - `iOS 15.0`
      - 其他说明
        - 支持 `iOS 15.2` ~ `15.4.1` 越狱可能性：有希望
          - 但是需要有人去完整修复 `Fugu15` ，才能利用该漏洞进行稳定地越狱
        - 另外：`iOS 15.5+` 越狱，基本没希望
  - 包管理器
    - 不支持：Cydia
      - `iOS 15` 更新了部分底层机制，导致了Cydia无法工作
        - 注：本身Cydia正在逐渐消亡， `Sileo` 等包管理器正在逐渐取代Cydia
    - 支持：`Sileo`
      - 也可以改为：`Zebra` 、 `Saily`

XinaA15越狱前

XinaA15的历史版本和下载地址

- XinaA15的下载来源
  - 官网
    - https://xina.ss03.cn/
  - Github
    - https://github.com/jacksight/xina520_official_jailbreak/releases/
- XinaA15的历史版本和下载地址
  - 工具版本： 1.1.8
    - 发布日期： 2023年03月13日
    - 更新内容： 修复一些已知问题，提升越狱稳定性。
    - 下载安装地址
      - ipa文件
        - http://apt.xina.vip/XinaA12.1.1.8.ipa
        - https://www.lanzouy.com/idV3Q0qvsr7a
        - https://github.com/jacksight/xina520_official_jailbreak/releases/download/v1.1.8/XinaA12.1.1.8.ipa
      - TrollStore在线安装URL地址
        - apple-magnifier://install?url=http://apt.xina.vip/XinaA12.1.1.8.ipa
  - 工具版本： 1.1.7.2
    - 发布日期： 2023年03月04日
    - 更新内容： 修复 dpkg LC_ RPATH Library/Frameworks 目录，修复越狱错误
    - 下载安装地址
      - ipa文件
        - https://apt.xina.vip/XinaA12.1.1.7.2.ipa
        - https://www.lanzouy.com/iSeyK0qvmqja
      - TrollStore在线安装URL地址
        - apple-magnifier://install?url=https://apt.xina.vip/XinaA12.1.1.7.2.ipa
  - 工具版本： 1.1.7.1
    - 发布日期： 2023年02月27日
    - 更新内容： 使用全新的UI设计，降低粉屏重启几率，调整了注销和软重启机制，提升越狱稳定性。
    - 下载安装地址
      - ipa文件
        - https://apt.xina.vip/XinaA12.1.1.7.1.ipa
        - https://www.lanzouy.com/ijDbt0qvmp4j
      - TrollStore在线安装URL地址
        - apple-magnifier://install?url=https://apt.xina.vip/XinaA12.1.1.7.1.ipa
  - 工具版本： 1.1.6.2
    - 发布日期： 2023年01月08日
    - 更新内容： 添加签名屏蔽注入功能，移除可选择管理器，修复一些已知问题，提升越狱稳定性。
    - 下载安装地址
      - ipa文件
        - https://apt.xina.vip/XinaA12.1.1.6.2.ipa
        - https://www.lanzouy.com/iXcgT0qvmntc
      - TrollStore在线安装URL地址
        - apple-magnifier://install?url=https://apt.xina.vip/XinaA12.1.1.6.2.ipa
  - 工具版本： 1.1.5
    - 发布日期： 2022年12月19日
    - 更新内容： 修复一些已知问题，提升系统稳定性，另外！Sileo软件包上出现35选项，不建议更新。
    - 下载安装地址
      - ipa文件
        - https://www.mediafire.com/file/o67m0jqchx2h2kp/XinaA12.1.1.5.ipa/file

        - https://www.lanzouy.com/i0FDG0qvmm4b
      - TrollStore在线安装URL地址
        - apple-magnifier://install?url=https://apt.xina.vip/XinaA12.1.1.5.ipa
  - 工具版本： 1.1.4.1
    - 发布日期： 2022年12月14日
    - 更新内容： 修复 iPad mini 6 安装插件不生效的问题。
    - 下载安装地址
      - ipa文件
        - https://www.lanzouy.com/iKgGi0inv7je
        - https://www.lanzouy.com/i0TQK0qvmi2f
  - 工具版本： 1.1.4
    - 发布日期： 2022年12月14日
    - 更新内容： 修复一些已知问题，添加 iPad mini 6 支持，提升系统稳定性。
    - 下载安装地址
      - ipa文件
        - https://github.com/jacksight/xina520_official_jailbreak/releases/download/v1.1.4/XinaA12.1.1.4.ipa
        - https://www.lanzouy.com/ifUE60qvmklg
  - 工具版本： 1.1.3.6
    - 发布日期： 2022年12月7日
    - 更新内容： 加入 iOS 15.0 - 15.1.1 A12 - A15/M1 系统支持，修复各种依赖的问题，提升稳定性。
    - 下载安装地址
      - ipa文件
        - https://raw.githubusercontent.com/jacksight/xina520_official_jailbreak/main/XinaA12.1.1.3.6.ipa
        - https://www.lanzouy.com/is4AS0qvmfmh

XinaA15越狱过程

此处介绍用 `XinaA15` 给 `iOS 15.1` 的 `A13` 的 `iPhone11` 越狱的过程：

准备设备

- 此处：咸鱼上买了个二手的（ `A13` 芯片）iPhone11
  - 满足前提
    - 芯片： `A12+`
      - 此处： `A13`
    - iOS版本： `iOS 15.0` ~ `iOS 15.1.1`
      - 此处： `15.1`
  - 图
    -


11:58
通用
关于本机
名称
iPhone11_151
软件版本
15.1
型号名称
iPhone 11
型号号码
MWN12CH/A
序列号
F4GCV58FN73V
歌曲
0
视频
0
照片
1
应用程序
1
总容量
64 GB
可用容量
52.13 GB
无线局域网地址
B4:40:A4:4D:4D:46
蓝牙
B4:40:A4:4A:48:7D
调制解调器固件
3.00.00
SEID

- 

XinaA15 给 iOS 15.1 的 iPhone11 越狱过程

概述

- XinaA15越狱步骤概述
  - 先去下载和安装TrollStore，包括配置和初始化TrollStore
    - 详见：
      - TrollStore · iOS逆向开发：iPhone越狱
  - 去XinaA15官网，下载最新版XinaA15的ipa文件
  - 用TrollStore安装XinaA15的app（ipa文件）

- 进入XinaA15的app中，点击：开启越狱
- 重启桌面，再次进入XinaA15的app，看到提示：当前已在越狱状态中，则表示：越狱成功

后续详细介绍整个过程。

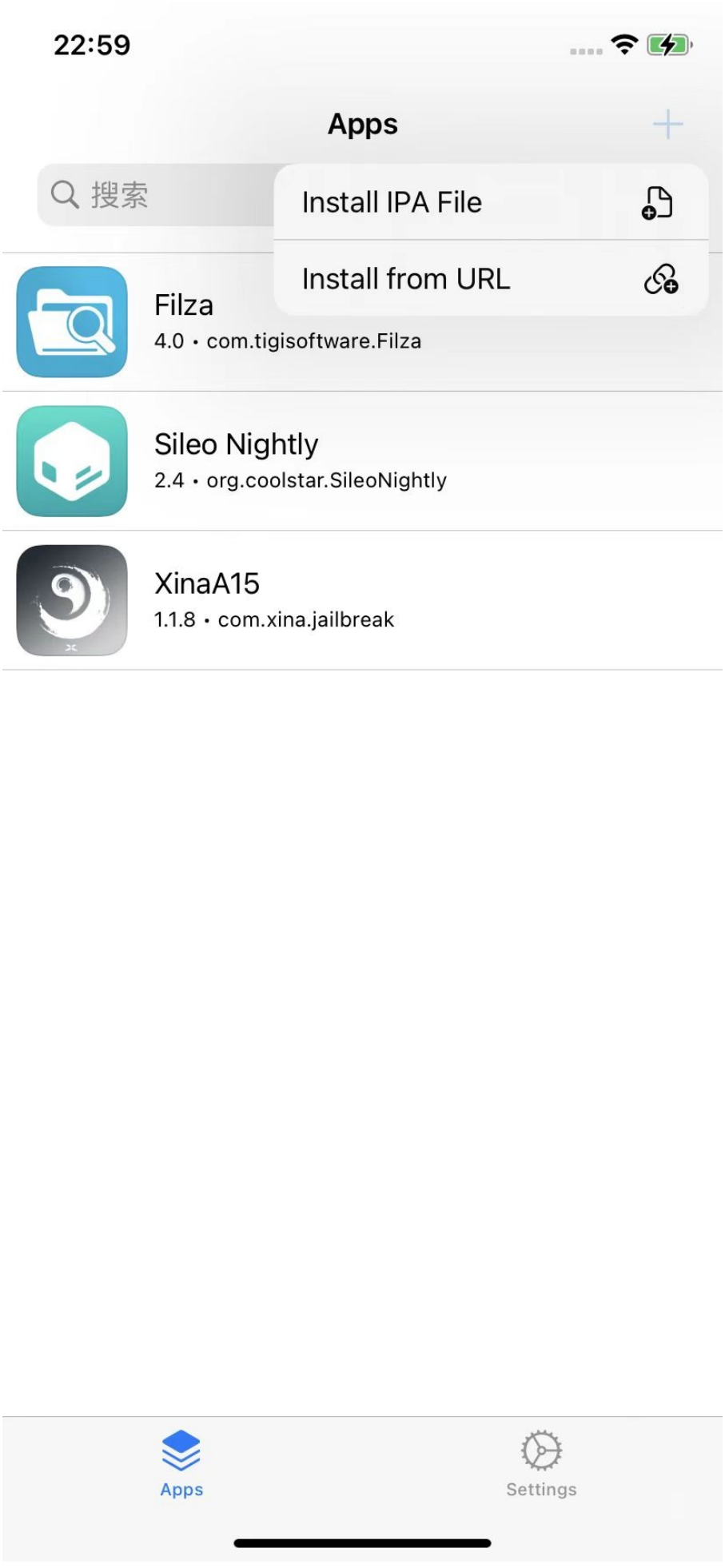
安装XinaA15

有多种方式去：

安装XinaA15的安装方式

（推荐）用TrollStore安装

- 用TrollStore安装XinaA15
  - （推荐）方式1
    - 下载XinaA15的ipa文件
    - 把ipa文件传到iPhone中，用TrollStore打开（即可自动安装XinaA15）
  - 方式2
    - 直接从URL安装
      - 步骤：TrollStore->右上角　加号->Install from URL
        -


22:59
Apps
搜索
Install IPA File
Install from URL
Filza
4.0 • com.tigisoftware.Filza
Sileo Nightly
2.4 • org.coolstar.SileoNightly
XinaA15
1.1.8 • com.xina.jailbreak
Apps
Settings

        - 说明：自己没试过，应该可行
        - 备注
            - TrollStore在线安装URL地址
                - 比如
                    - 最新版 1.1.8的：
                        - `apple-magnifier://install?url=http://apt.xina.vip/XinaA12.1.1.8.ipa`

从Safari浏览器安装

- 从Safari浏览器安装
    - 比如地址：
        - https://iexmo.com/apps/install/jb-updated/xinajb.php
    - 说明：自己没去试
        - 估计即使可以安装，安装出来的也是旧版XinaA15，不是最新版本
            - 所以也不太推荐此方式

通过SideLoadly安装

- 通过SideLoadly
    - 图
        - 

    - 说明：自己没试过。估计可行

通过包管理器从源中安装

- 通过包管理器(Sileo等)直接添加XinaA15的源去安装
    - XinaA15的源地址
        - https://apt.xina.vip
    - 说明：貌似不是最新版，所以不是很推荐这种方式

下载XinaA15的ipa再用TrollStore去安装

下面就对于前面提到的：

用TrollStore安装XinaA15 中的方式1，即：

- 下载XinaA15的ipa文件
- 把ipa文件传到iPhone中，用TrollStore打开（即可自动安装XinaA15）

进行详细介绍。

下载XinaA的ipa文件

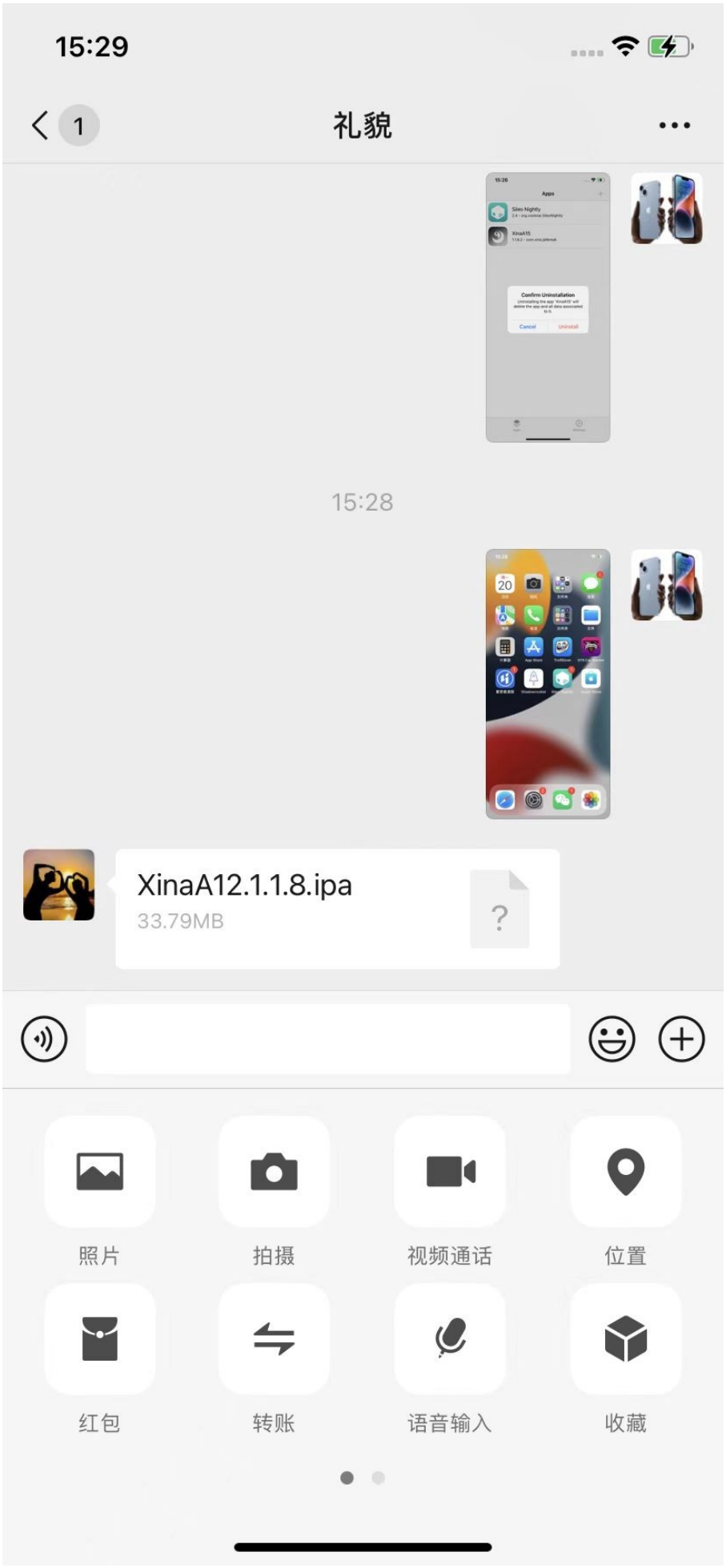
- 下载XinaA的ipa文件
  - 从官网 https://xina.ss03.cn/ 下载到最新版本的XinaA15的ipa文件
    - 比如：1.1.8 的 http://apt.xina.vip/XinaA12.1.1.8.ipa
      - 注意：此处直接下载没速度，最后改用迅雷，可以顺利下载

如何把XinaA15的ipa文件，传输到iPhone中

把XinaA15的ipa文件传输到iPhone中的多种方式

（推荐）用微信传输

- 支持场景：越狱前/越狱后
- 具体逻辑：Mac中和iPhone中都登录（同一账号的微信，或其他微信朋友，总之确保能传输文件）
- 步骤=效果
  - 从Mac中把ipa传到微信中
    -


15:29
1
礼貌
15:28
XinaA12.1.1.8.ipa
33.79MB
照片
拍摄
视频通话
位置
红包
转账
语音输入
收藏

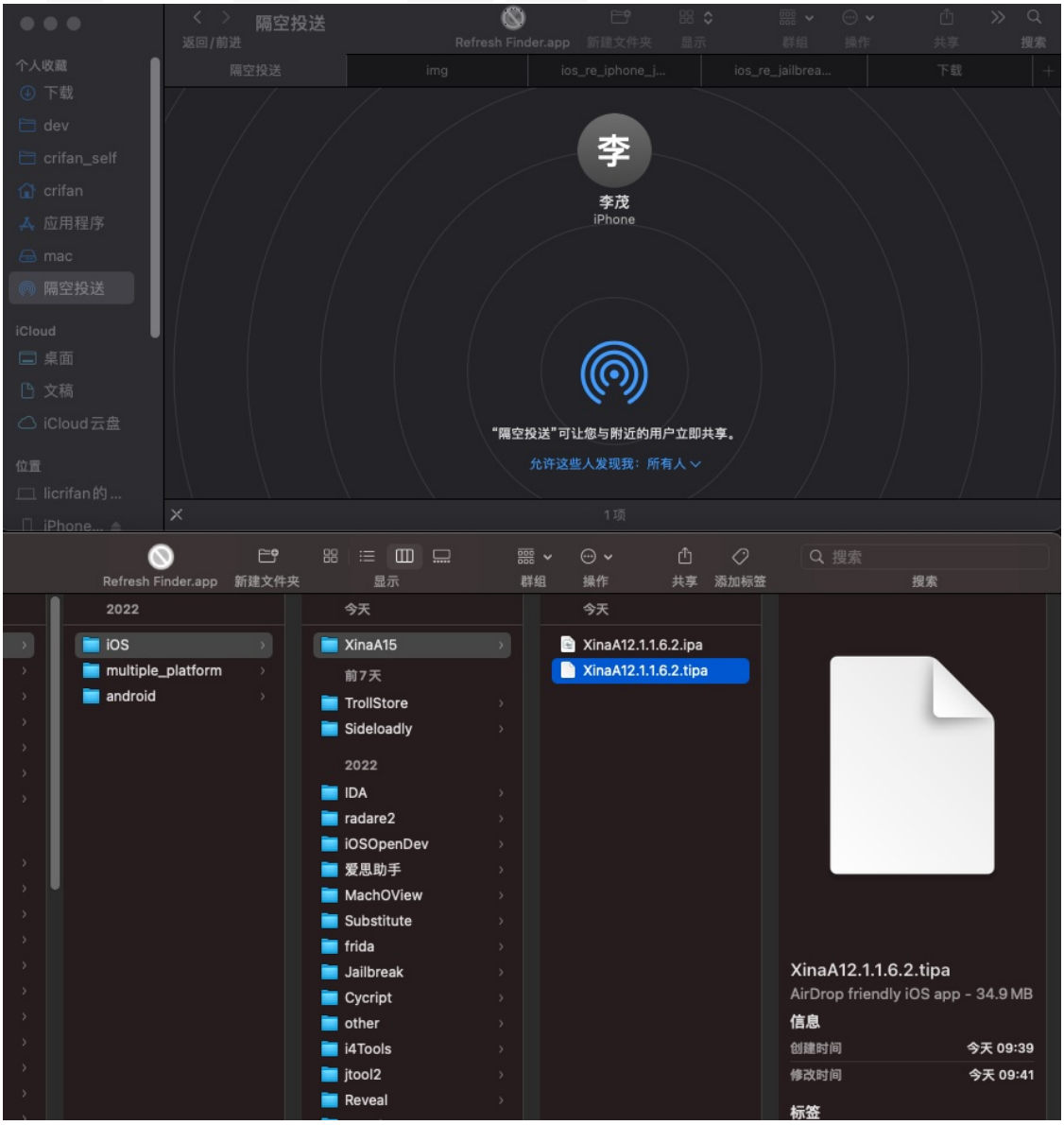
- 点击进入，点击：用其他应用打开
  -


15:29
?
XinaA12.1.1.8.ipa
文件大小: 33MB
微信暂不可以打开此类文件，你可以使用其他应用打开并预览。
用其他应用打开

- 点击：`TrollStore`
  -


15:30
XinaA12.1.1.8.ipa
文件大小: 33MB
微信暂不可以打开此类文件，你可以使用其
XinaA12.1.1.8
iOS App · 35.4 MB
隔空投送
TrollStore
邮件
微信
拷贝
存储到“文件”
编辑操作...

通过AirDrop隔空投送传输tipa文件

- 支持场景：越狱前/越狱后
- 主要逻辑：把 `.ipa` 改为 `.tipa`，通过 `AirDrop` = `隔空投送` 发送到 `iPhone` 中，在用 `TrollStore` 打开
- 具体步骤
  - 把之前的 `.ipa` 文件改后缀为 `.tipa`，然后通过 `AirDrop` = `隔空投送` 发送到iPhone中
    - 

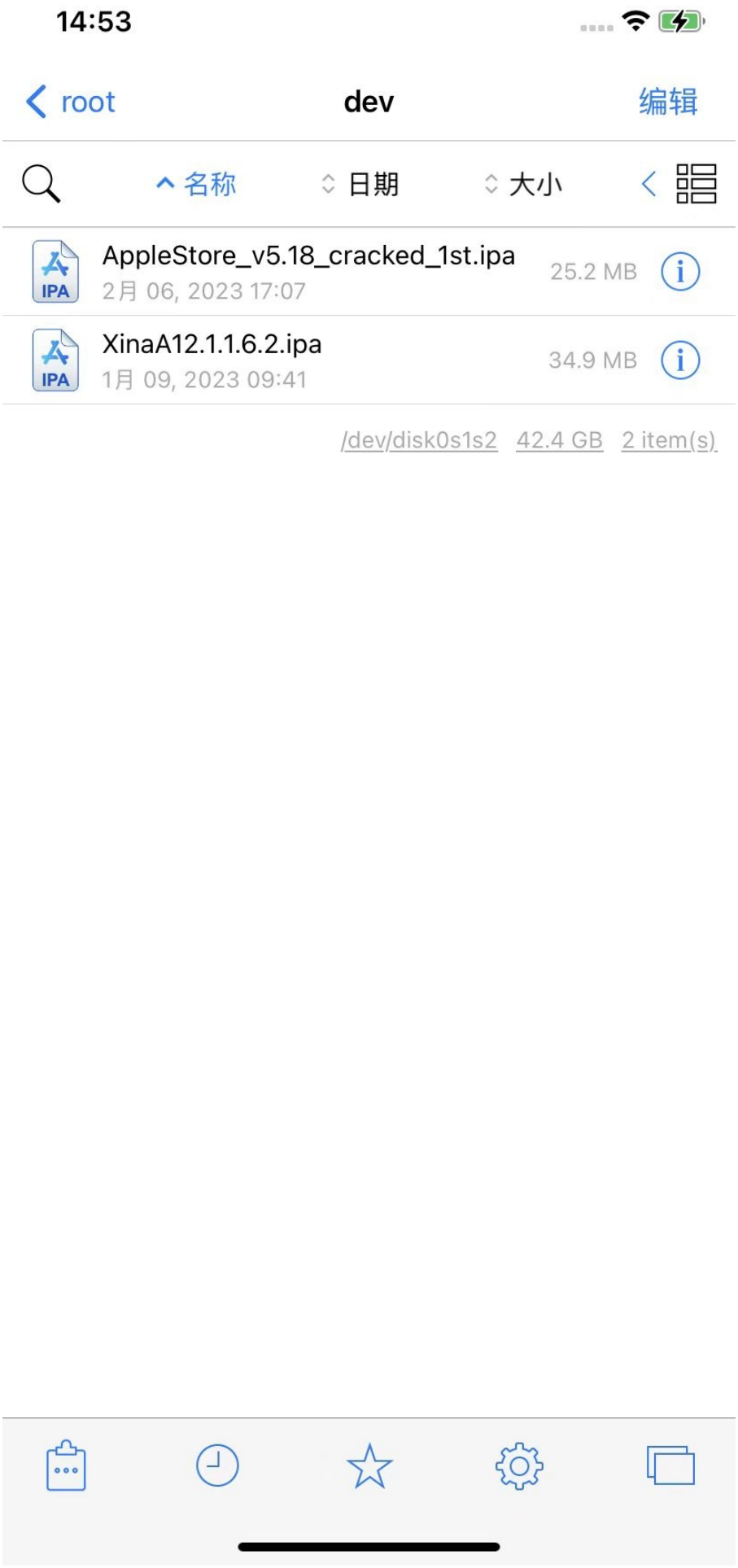
  - iPhone中会自动出现弹框：打开方式，选择 `TrollStore`
    -


10:32
天气
查找
快捷指令
家庭
通讯录
文件
股市
翻译
隔空投送
打开方式...
TrollStore
文件
取消
电话
GTA Car Tracker
TrollStore
Shadowrocket

- 额外说明：此处有点诡异的是，Mac直接传送ipa到iPhone，但最后始终无法顺利保存到此处的 `iCloud` 中
  - 也就无法实现网上很多人说的，TrollStore从iCloud中安装ipa的方式了

通过scp拷贝到iPhone中，再用Filza去打开

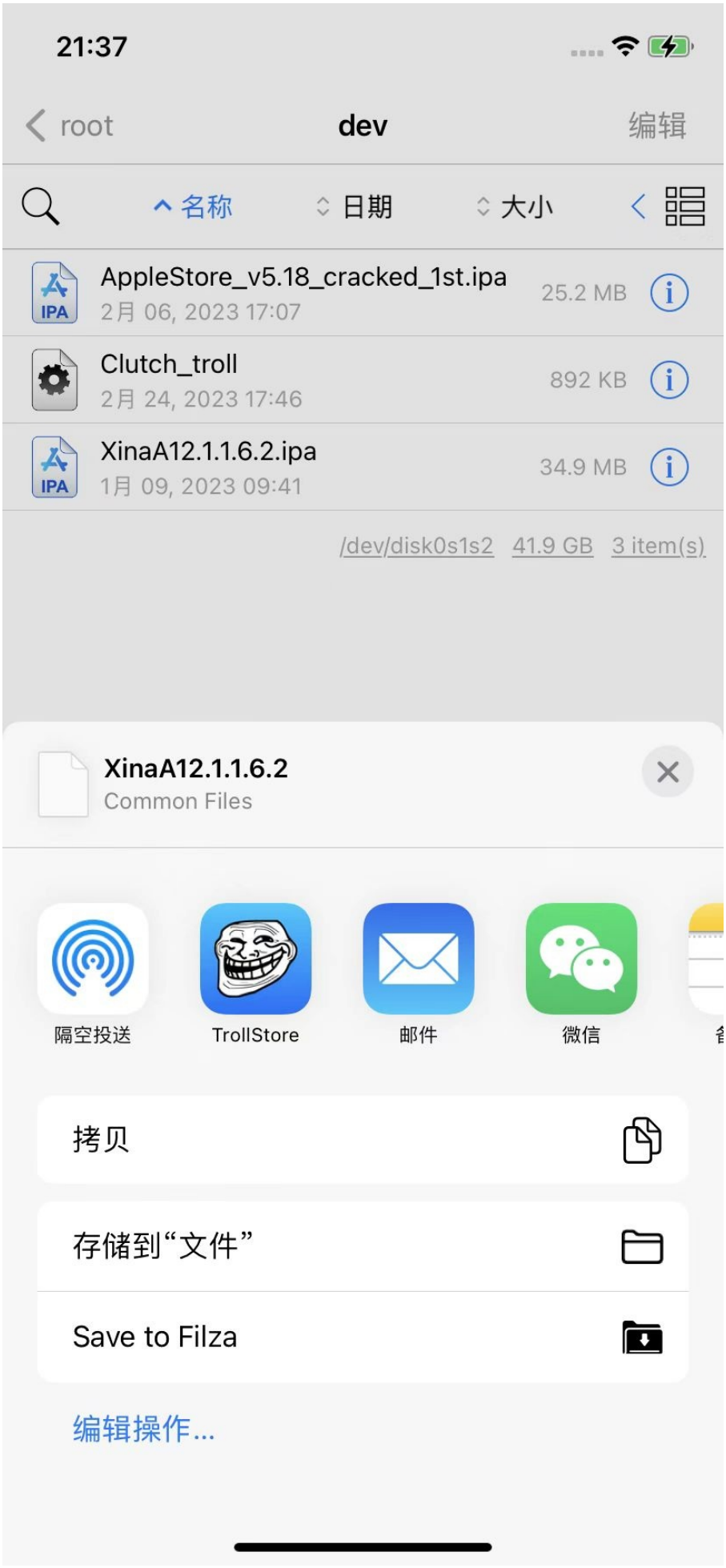
- 支持场景：越狱后
- 步骤：
  - 通过scp拷贝到iPhone中
    - `scp -p XinaA12.1.1.6.2.ipa root@192.168.2.12:/var/root/dev/XinaA12.1.1.6.2.ipa`
  - 再用Filza去打开
    - Filza中的XinaA15的ipa文件
      -


14:53
root
dev
编辑
名称
日期
大小
AppleStore_v5.18_cracked_1st.ipa
2月 06, 2023 17:07
25.2 MB
XinaA12.1.1.6.2.ipa
1月 09, 2023 09:41
34.9 MB
/dev/disk0s1s2
42.4 GB
2 item(s)

- 长按-》右键菜单
  -


21:37
XinaA12.1.1.6.2.ipa
1月 09, 2023 09:41
34.9 MB
XinaA12.1.1.6.2.ipa
选择
复制
移动
重命名
回收站
使用 App打开
打开方式
压缩为 ZIP 文件
属性
添加
Scripts

- 继续安装
  - 方式1：选择 `使用App打开` -> 选择 `TrollStore`
    -

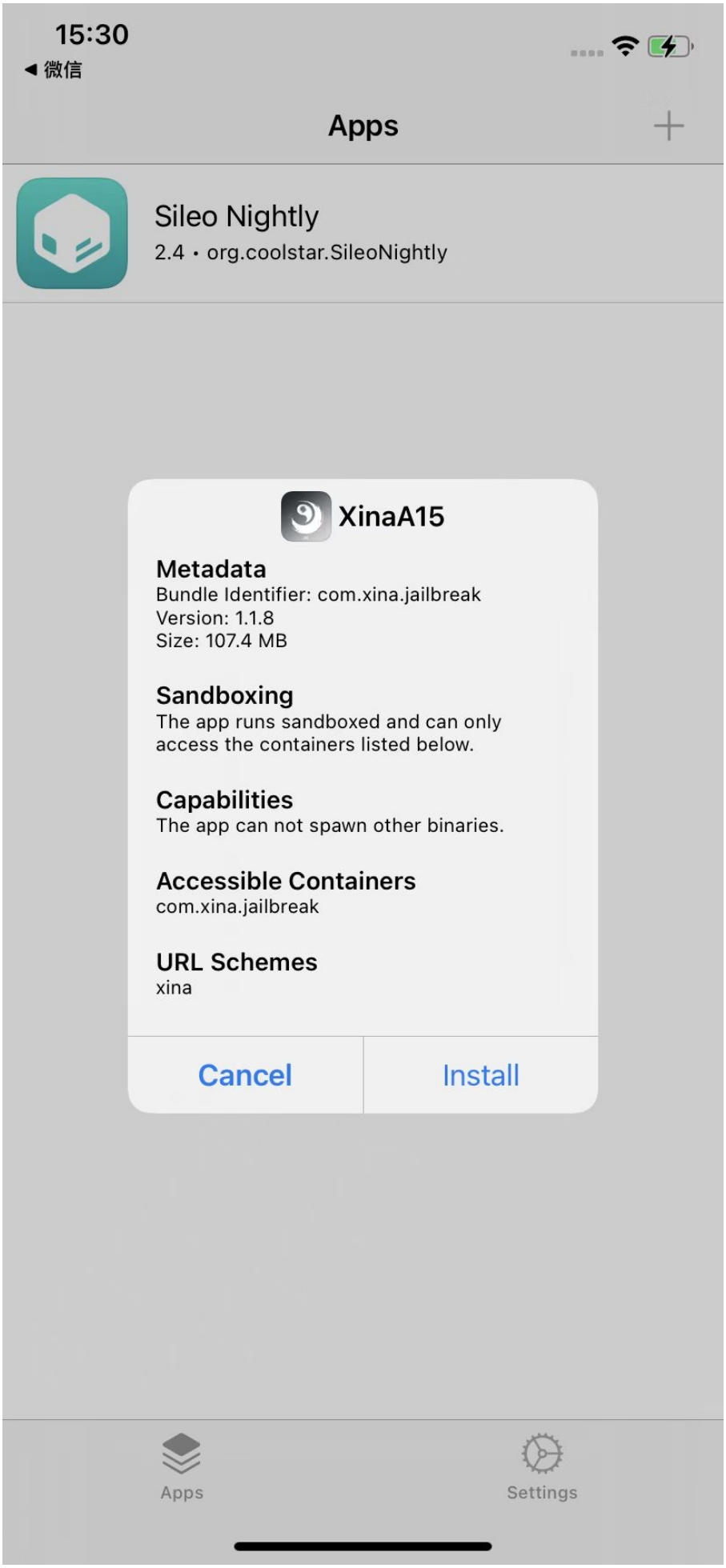

21:37
root
dev
编辑
名称
日期
大小
AppleStore_v5.18_cracked_1st.ipa
2月 06, 2023 17:07
25.2 MB
Clutch_troll
2月 24, 2023 17:46
892 KB
XinaA12.1.1.6.2.ipa
1月 09, 2023 09:41
34.9 MB
/dev/disk0s1s2 41.9 GB 3 item(s)
XinaA12.1.1.6.2
Common Files
隔空投送
TrollStore
邮件
微信
拷贝
存储到“文件”
Save to Filza
编辑操作...

- 方式2：选择 `打开方式` -> 选择 `TrollStore`
  -

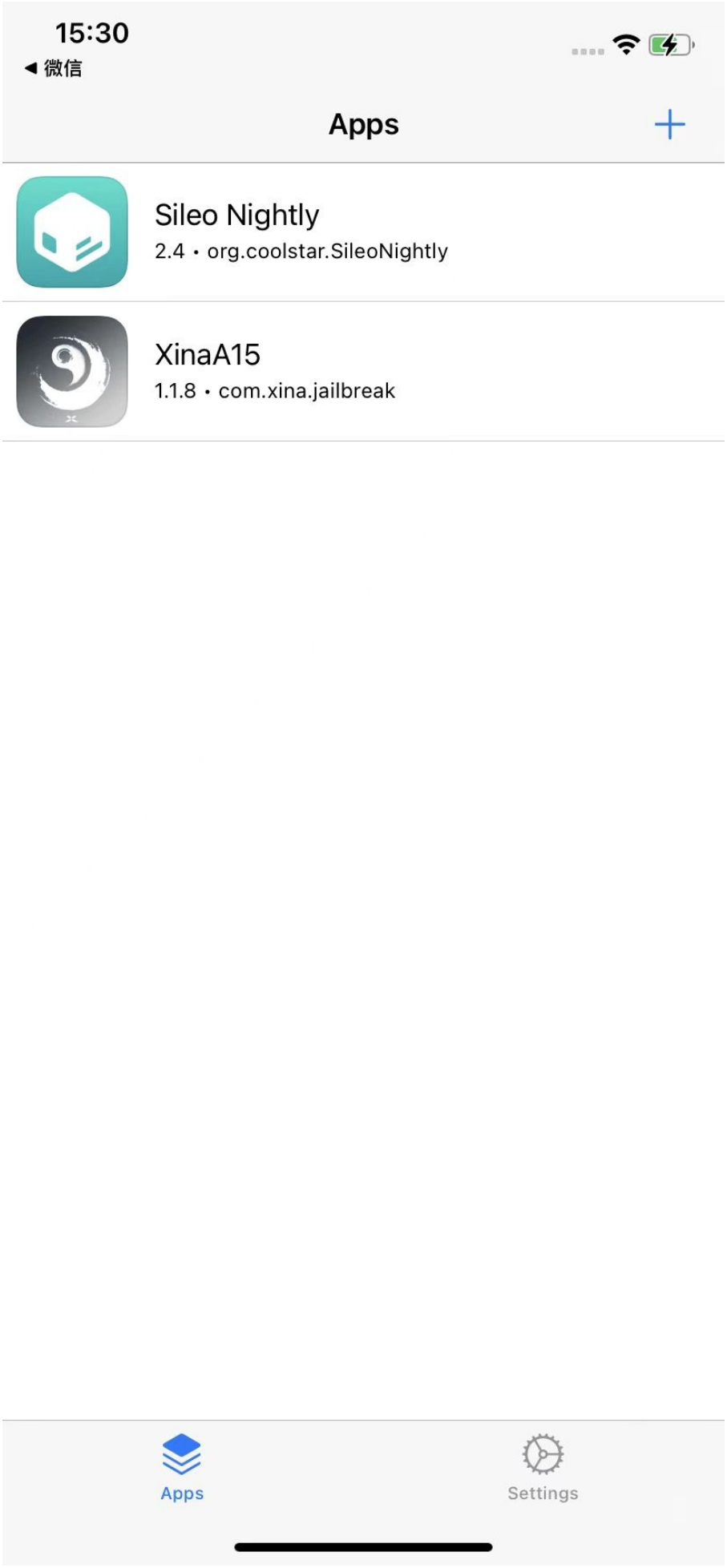

21:37
root
dev
编辑
名称
日期
大小
AppleStore_v5.18_cracked_1st.ipa
2月 06, 2023 17:07
25.2 MB
Clutch_troll
2月 24, 2023 17:46
892 KB
XinaA12.1.1.6.2.ipa
1月 09, 2023 09:41
34.9 MB
/dev/disk0s1s2 41.9 GB 3 item(s)
打开方式 ...
TrollStore
XinaA15
爱思极速版
Shadowrocket
微信
图片浏览器
快速查看
Web Viewer
属性表编辑器
文本编辑器
取消

用TrollStore打开并自动安装XinaA15的ipa文件

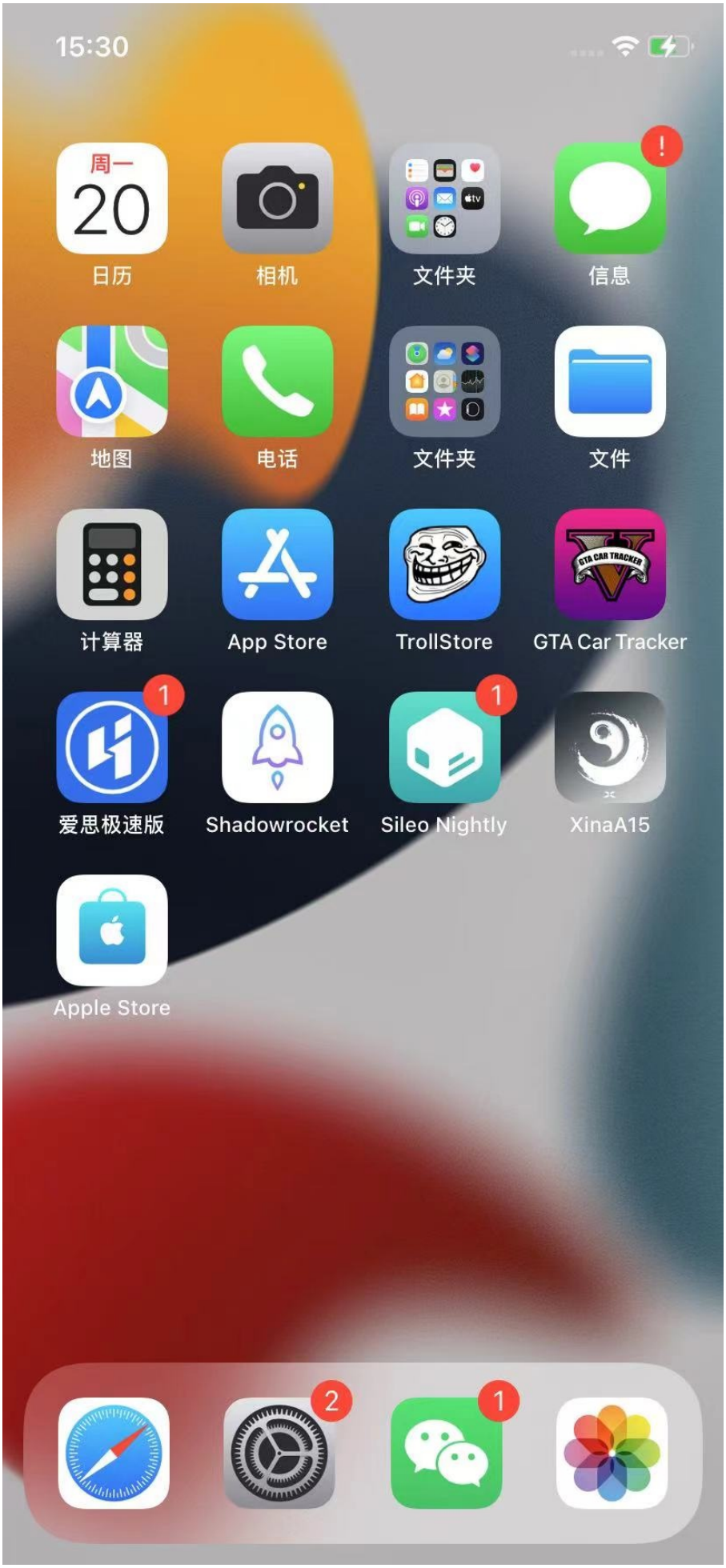
- 前提：已把XinaA15的ipa文件，传输到iPhone中，且已选择用TrollStore去打开
- 后续步骤：用TrollStore打开XinaA15的ipa并自动安装的过程
  - 会自动出现弹框，点击 `Install`
    -


15:30
◀ 微信
Apps
Sileo Nightly
2.4 • org.coolstar.SileoNightly
XinaA15
Metadata
Bundle Identifier: com.xina.jailbreak
Version: 1.1.8
Size: 107.4 MB
Sandboxing
The app runs sandboxed and can only access the containers listed below.
Capabilities
The app can not spawn other binaries.
Accessible Containers
com.xina.jailbreak
URL Schemes
xina
Cancel
Install
Apps
Settings

- 显示Installing
- 安装完毕后，即可在 `Apps` tab页面中看到已安装的 `XinaA15`
  -


15:30
◀ 微信
Apps
Sileo Nightly
2.4 • org.coolstar.SileoNightly
XinaA15
1.1.8 • com.xina.jailbreak
Apps
Settings

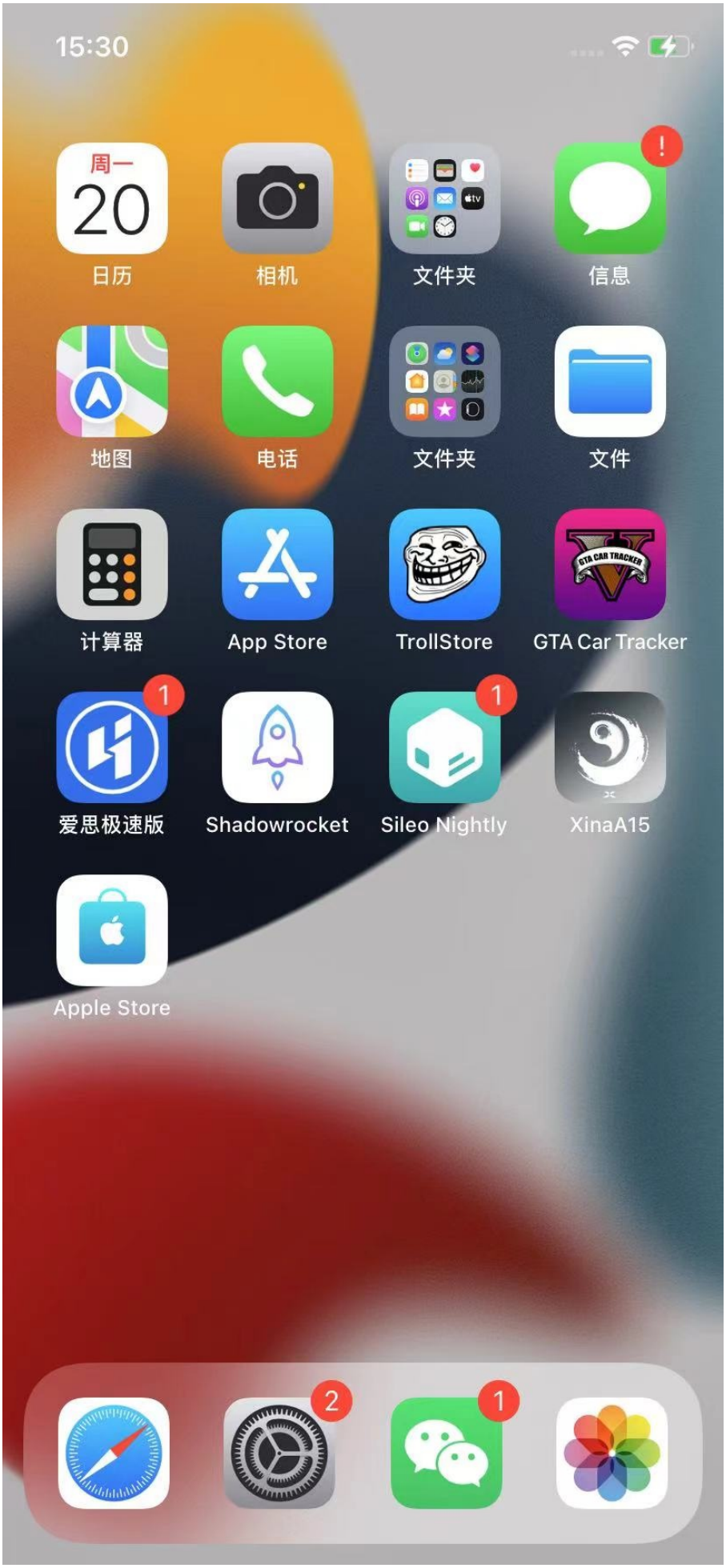
- 安装后的桌面出现XinaA15的图标了
  -


15:30
周一
20
日历
相机
文件夹
信息
地图
电话
文件夹
文件
计算器
App Store
TrollStore
GTA Car Tracker
爱思极速版
Shadowrocket
Sileo Nightly
XinaA15
Apple Store

用XinaA15越狱=开启越狱

此处介绍用 `XinaA15` 给 `iOS 15.1` 的 `iPhone11` 越狱的过程：

- 点击iPhone桌面的XinaA15的app图标：
  -


15:30
周一
20
日历
相机
文件夹
信息
地图
电话
文件夹
文件
计算器
App Store
TrollStore
GTA Car Tracker
爱思极速版
Shadowrocket
Sileo Nightly
XinaA15
Apple Store

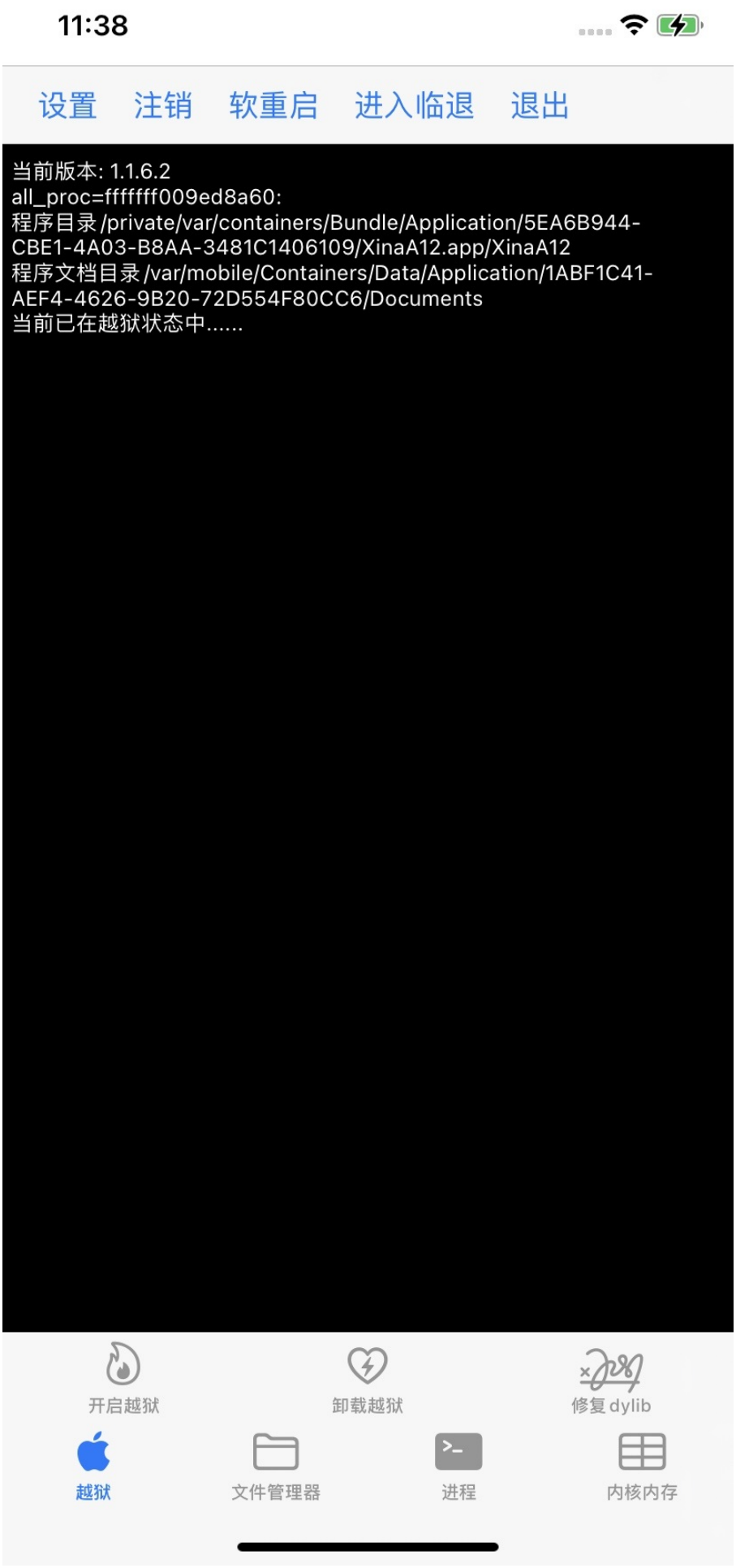
- 进入XinaA15的app的主页面，点击：`开启越狱`
  -


11:33
设置
注销
软重启
进入临退
退出
注意
Disclaimer: This version is a voluntary test version!...
该版本功劳属于你们
@LinusHenze @jaakerblom @zhuowei @tihmstar
@CStar_AD @Jakeashacks @opa334dev
语言 (Lang...
中文
EN
自动启动 ssh 服务
重新安装越狱环境
手动注入模式 (暂不使用)
安全模试 (出问题用)
当前版本: 1.1.6.2
开启越狱
卸载越狱
修复 dylib
越狱
文件管理器
进程
内核内存

- 接着会输出很多log日志
  -


11:33
设置
注销
软重启
进入临退
退出
3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B7
62F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus//usr/
bin/zfgrep
准备修复 link /private/preboot/
3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B7
62F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus/bin/
zforce
link 指向文件已修复 0 /usr/bin/zforce /private/preboot/
3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B7
62F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus//usr/
bin/zforce
准备修复 link /private/preboot/
3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B7
62F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus/bin/
gunzip
link 指向文件已修复 0 /usr/bin/gunzip /private/preboot/
3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B7
62F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus//usr/
bin/gunzip
准备修复 link /private/preboot/
3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B7
62F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus/sbin/
halt
link 指向文件已修复 0 /usr/sbin/reboot /private/preboot/
3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B7
62F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus//usr/
sbin/reboot
准备修复 link /private/preboot/
3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B7
62F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus/etc/
ssl/cert.pem
准备修复 link /private/preboot/
3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B7
62F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus/etc/
dpkg/origins/default
jailbreakd_safe 服务已启动...
启动 jailbreakd 服务....
jailbreakd 服务已启动...
启动 launchdhook 服务....
launchctl load com.openssh.sshd.plist...
签名 debugserver...
签名 xpcproxy...
签名 launchctl...
永久签名自己...
准备重启用户空间请稍等...
开启越狱
卸载越狱
修复 dylib
越狱
文件管理器
进程
内核内存

- 会重启桌面=respring
- 重启后，再次点击进入XinaA15的app，即可看到主页中提示：`当前已在越狱状态中`，表示：已越狱成功
  -

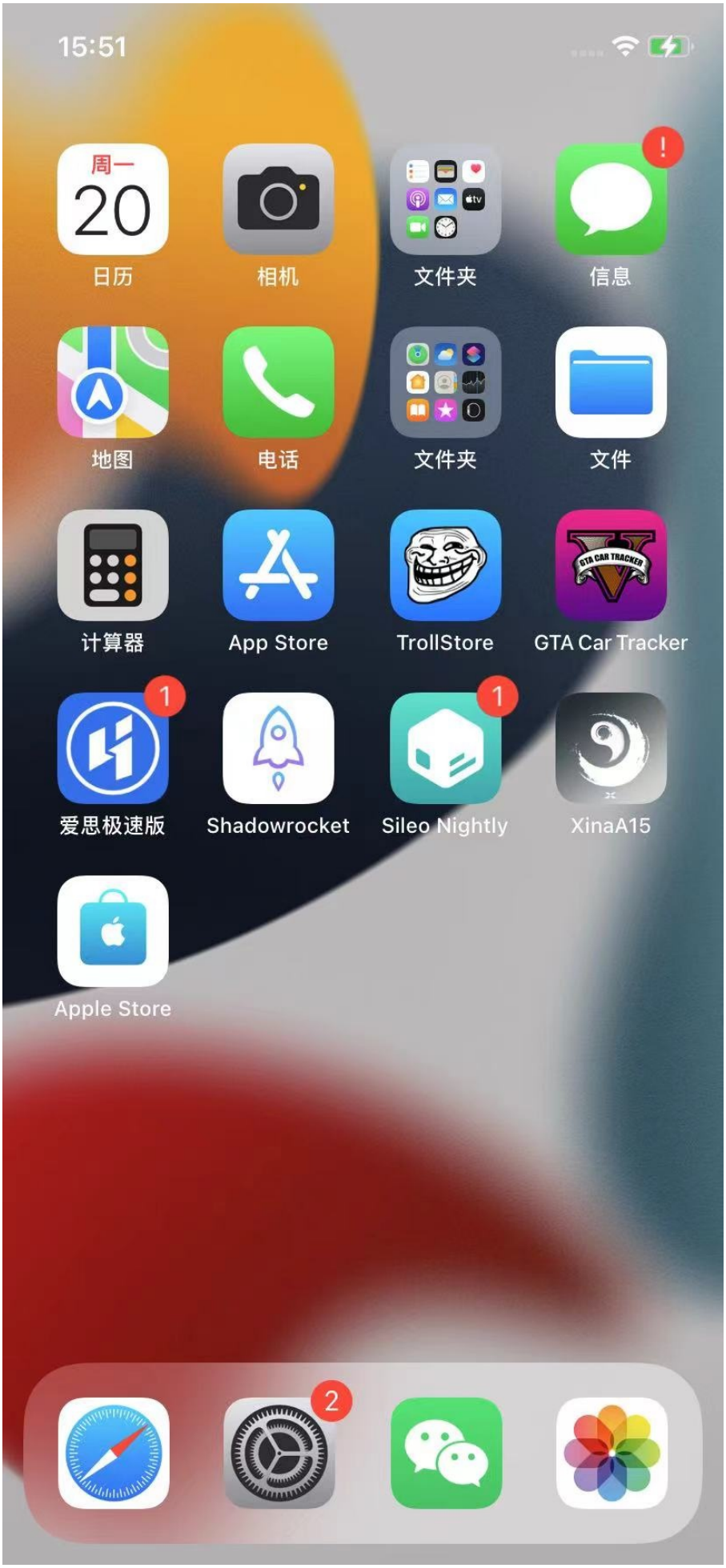

11:38
设置
注销
软重启
进入临退
退出
当前版本: 1.1.6.2
all_proc=ffffffff009ed8a60:
程序目录/private/var/containers/Bundle/Application/5EA6B944-CBE1-4A03-B8AA-3481C1406109/XinaA12.app/XinaA12
程序文档目录/var/mobile/Containers/Data/Application/1ABF1C41-AEF4-4626-9B20-72D554F80CC6/Documents
当前已在越狱状态中.....
开启越狱
卸载越狱
修复 dylib
越狱
文件管理器
进程
内核内存

XinaA15越狱后

基本信息

桌面图标和app

- XinaA15越狱后的，典型的桌面图标和app：
  - XinaA15
  - Sileo Nightly
  - TrollStore
  - GTA Car Tracker
- 图
  -


15:51
周一
20
日历
相机
文件夹
信息
地图
电话
文件夹
文件
计算器
App Store
TrollStore
GTA Car Tracker
爱思极速版
Shadowrocket
Sileo Nightly
XinaA15
Apple Store

越狱后iPhone信息

- 爱思助手查看iPhone的状态，可以显示：`已越狱`
  - 

（XinaA15越狱后）如何使用

- （XinaA15越狱后）如何使用
  - 就是和普通的越狱iPhone一样去使用即可
    - 用ssh连接和操作iPhone
      - 其中已自带 `OpenSSH` 的 `ssh` 的server和client，可以直接使用ssh了（默认的ssh的用户名：`root`，密码：`alpine`）
    - 用包管理器 `Sileo` 安装各种插件
      - 注：此处是专门适配的特殊版本的 `Sileo Nightly`
    - 用各种越狱插件工具 `Logos` / `Theos` 、 `iOSOpenDev` 、 `MonkeyDev` 等去开发越狱插件
    - `frida` ：截至 `20230406` ，不可用
      - 可以安装frida，但是Mac中使用frida的命令（比如 `frida-ps -U` ）会导致iPhone重启，而无法正常使用
    - 等等

越狱后的功能和插件使用

详解XinaA15中如何使用ssh

- 详解XinaA15中如何使用ssh
  - 说明：XinaA15越狱后（的iOS 15.1的iPhone11中）默认已开启ssh服务
    - 配置中显示了：`自动启动ssh服务`
      -


11:39
设置
注销
软重启
进入临退
退出
注意
Disclaimer: This version is a voluntary test version!...
该版本功劳属于你们
@LinusHenze @jaakerblom @zhuowei @tihmstar
@CStar_AD @Jakeashacks @opa334dev
语言 (Lang...
中文
EN
自动启动 ssh 服务
重新安装越狱环境
手动注入模式 (暂不使用)
安全模试 (出问题用)
当前版本: 1.1.6.2
开启越狱
卸载越狱
修复 dylib
越狱
文件管理器
进程
内核内存

- 然后：直接用其他客户端去通过ssh连接iPhone即可
  - ssh（server）默认配置
    - 端口：`22`
    - 用户名：`root`
    - 密码：`alpine`
  - 客户端
    - 比如
      - Mac中默认的终端Terminal
  - 命令
    - `ssh root@192.168.2.12`
      - 其中：`192.168.2.12` 是当前越狱iPhone的IP地址
  - 注：
    - 免密登录
      - `ssh-copy-id root@192.168.2.12`

frida：不可用

- 关于frida：不可用
  - 现象：Mac中用 `frida-ps -U`，就会导致iPhone重启
  - 当前版本
    - XinaA15 最新版 `1.1.8`
    - （iPhone11 + Mac中都是）Frida 最新版 `16.0.11`

其他越狱插件支持情况

- XinaA15是rootless越狱：之前很多插件会不可用，或者有问题
  - 网上有人整理了各种插件对于XinaA15的支持状态
    - 详见：【整理】XinaA15适配支持的越狱插件

XinaA15本身的使用

（重启而丢失越狱后如何）恢复越狱

- 恢复越狱
  - 现象
    - iPhone重启后，丢失越狱，具体现象：
      - XinaA15：没有显示 当前已在越狱状态中
        -


17:14
中文
EN
Xina iOS15 越狱工具
by Xina520
For iOS15.0 – iOS15.1.1
UI by Alierya
免责声明:使用该软件造成一切后果由自己承担!
Process directory=/private/var/containers/Bundle/Application/DF6A47F9-E725-4EDE-AA94-A00CB9EC256C/XinaA12.app/XinaA12
Program document directory=/var/mobile/Containers/Data/Application/1832D574-D4E9-4C89-98FC-72E187E37988/Library
注销
软重启
临退
开启越狱
当前版本: 1.1.8

        - 点击包管理器Sileo Nightly，会崩溃闪退而无法打开
        - ssh也无法使用
- 核心步骤
    - 保持默认配置，直接点击 开启越狱 即可
        - 默认配置指的是：
            - 勾选了
                - 自动启动ssh服务
                - 首次自动安装巨魔
            - 没勾选
                - 重新安装越狱环境
                - 安全模式（有问题再开启）
        - 恢复越狱后
            - 显示：当前已在越狱状态中
                -


17:05
中文
EN
Xina iOS15 越狱工具
by Xina520
For iOS15.0 – iOS15.1.1
UI by Alierya
免责声明:使用该软件造成一切后果由自己承担!
get kernelbase=fffffff00f094000;
get kernelslide=8090000;
get allproc=fffffff009ed8a60;
uid=0
Process directory=/private/var/containers/Bundle/Application/DF6A47F9-E725-4EDE-AA94-A00CB9EC256C/XinaA12.app/XinaA12
Program document directory=/var/mobile/Containers/Data/Application/1832D574-D4E9-4C89-98FC-72E187E37988/Library
当前已在越狱状态中.....
注销
软重启
临退
重新越狱
当前版本: 1.1.8

        - 注意
            - 如果点击开启越狱后XinaA15崩溃，就多试几次

重新安装越狱

- 背景：有时候出现越狱问题，而无法解决时，可以考虑，重新越狱
- 重新安装越狱
    - 重新恢复最初的越狱环境：
        - 核心步骤：XinaA15-》勾选：重新安装越狱环境-》点击：开启越狱 -》iPhone无需（不会）重启，最后输出log是：Done，即表明恢复越狱成功
        - 详细步骤：
            - XinaA15-》勾选：重新安装越狱环境 -》点击：开启越狱
                -


14:56
设置
注销
软重启
进入临退
退出
注意
Disclaimer: This version is a voluntary test version!...
该版本功劳属于你们
@LinusHenze @jaakerblom @zhuowei @tihmstar
@CStar_AD @Jakeashacks @opa334dev
语言 (Lang...
中文
EN
自动启动 ssh 服务
重新安装越狱环境
手动注入模式 (暂不使用)
安全模试 (出问题用)
当前版本: 1.1.6.2
开启越狱
卸载越狱
修复 dylib
越狱
文件管理器
进程
内核内存

- -》iPhone无需（不会）重启，最后输出log是：Done
  -


15:08
设置
注销
软重启
进入临退
退出
62F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus//usr/bin/zfgrep
准备修复 link /private/preboot/3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B762F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus/bin/zforce
link 指向文件已修复 0 /usr/bin/zforce /private/preboot/3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B762F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus//usr/bin/zforce
准备修复 link /private/preboot/3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B762F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus/bin/gunzip
link 指向文件已修复 0 /usr/bin/gunzip /private/preboot/3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B762F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus//usr/bin/gunzip
准备修复 link /private/preboot/3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B762F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus/sbin/halt
link 指向文件已修复 0 /usr/sbin/reboot /private/preboot/3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B762F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus//usr/sbin/reboot
准备修复 link /private/preboot/3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B762F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus/etc/ssl/cert.pem
准备修复 link /private/preboot/3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B762F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus/etc/dpkg/origins/default
jailbreakd_safe 已存在!
重启 jailbreakd
jailbreakd 服务已启动...
launchdhook 已存在! 重启 launchdhook 服务..
launchctl load com.openssh.sshd.plist...
签名 debugserver...
签名 xpcproxy...
签名 launchctl...
无需签名...
当前版本支持字体...
Done
开启越狱
卸载越狱
修复 dylib
越狱
文件管理器
进程
内核内存

        - 即表明恢复越狱成功

升级新版后，重新越狱

- 场景：比如之前的XinaA15的版本是 `1.1.6.2` ，想要升级到当前最新版本 `1.1.8`
- 升级新版后，重新越狱
    - 核心步骤：
        - 安装新版XinaA15的ipa
        - 勾选：重新安装越狱环境，再点击 开启越狱
            -


15:33
设置
部分功能请谨慎使用
文件管理
进程查看
内核查看
修复 dylib
自动启动 ssh 服务
重新安装越狱环境
安全模式 (有问题再开启)
首次自动安装巨魔
卸载程序
查看更新日志
更新下载连接
鸣谢:该工具功劳属于你们
@CStar_AD @Jakeashacks @opa334dev
@LinusHenze @jaakerblom @zhuowei @tihmstar

卸载XinaA15

- 卸载XinaA15
  - 点击：卸载越狱=UnJailbreak
    -


21:28
设置
注销
软重启
进入临退
退出
注意
Disclaimer: This version is a voluntary test version!...
该版本功劳属于你们
@LinusHenze @jaakerblom @zhuowei @tihmstar
@CStar_AD @Jakeashacks @opa334dev
语言 (Lang...
中文
EN
自动启动 ssh 服务
重新安装越狱环境
手动注入模式 (暂不使用)
安全模试 (出问题用)
当前版本: 1.1.6.2
开启越狱
卸载越狱
修复 dylib
越狱
文件管理器
进程
内核内存

- 弹框点击确定
  -


21:30
设置
注销
软重启
进入临退
退出
当前版本: 1.1.6.2
程序目录/private/var/containers/Bundle/Application/5EA6B944-CBE1-4A03-B8AA-3481C1406109/XinaA12.app/XinaA12
程序文档目录/var/mobile/Containers/Data/Application/1ABF1C41-AEF4-4626-9B20-72D554F80CC6/Documents
卸载需要在越狱的环境下进行
是否现在卸载所有越狱相关一切?
取消
确定
开启越狱
卸载越狱
修复 dylib
越狱
文件管理器
进程
内核内存

    - iPhone会重启，之后即可完成卸载。
- 注意
    - 如果此处（之前重启iPhone而导致的）已丢失越狱
        - 需要先恢复越狱：XinaA15中，点击 `开启越狱`
        - 然后才能继续卸载XinaA15

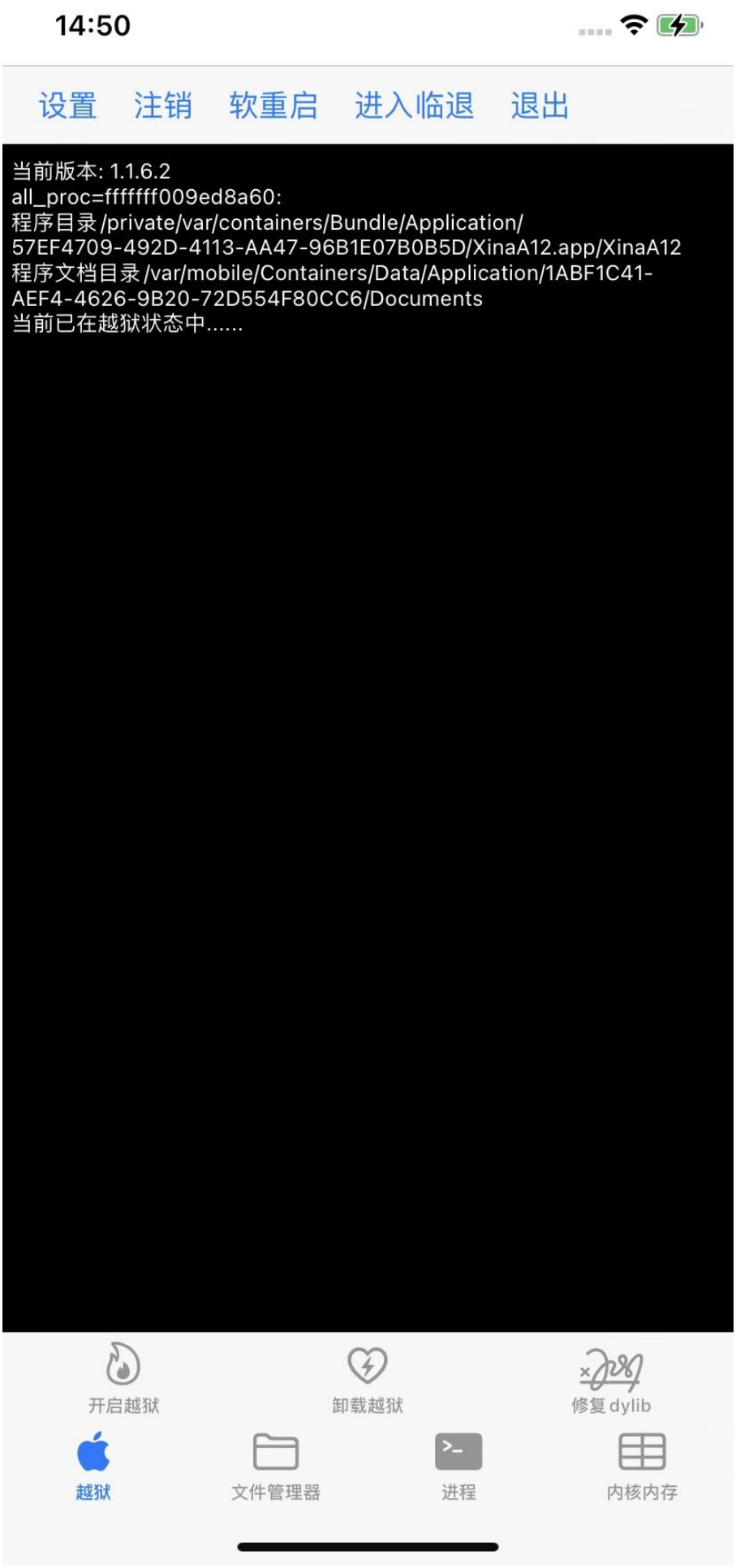
XinaA15的界面和功能

XinaA15 1.1.8之前：旧的UI界面

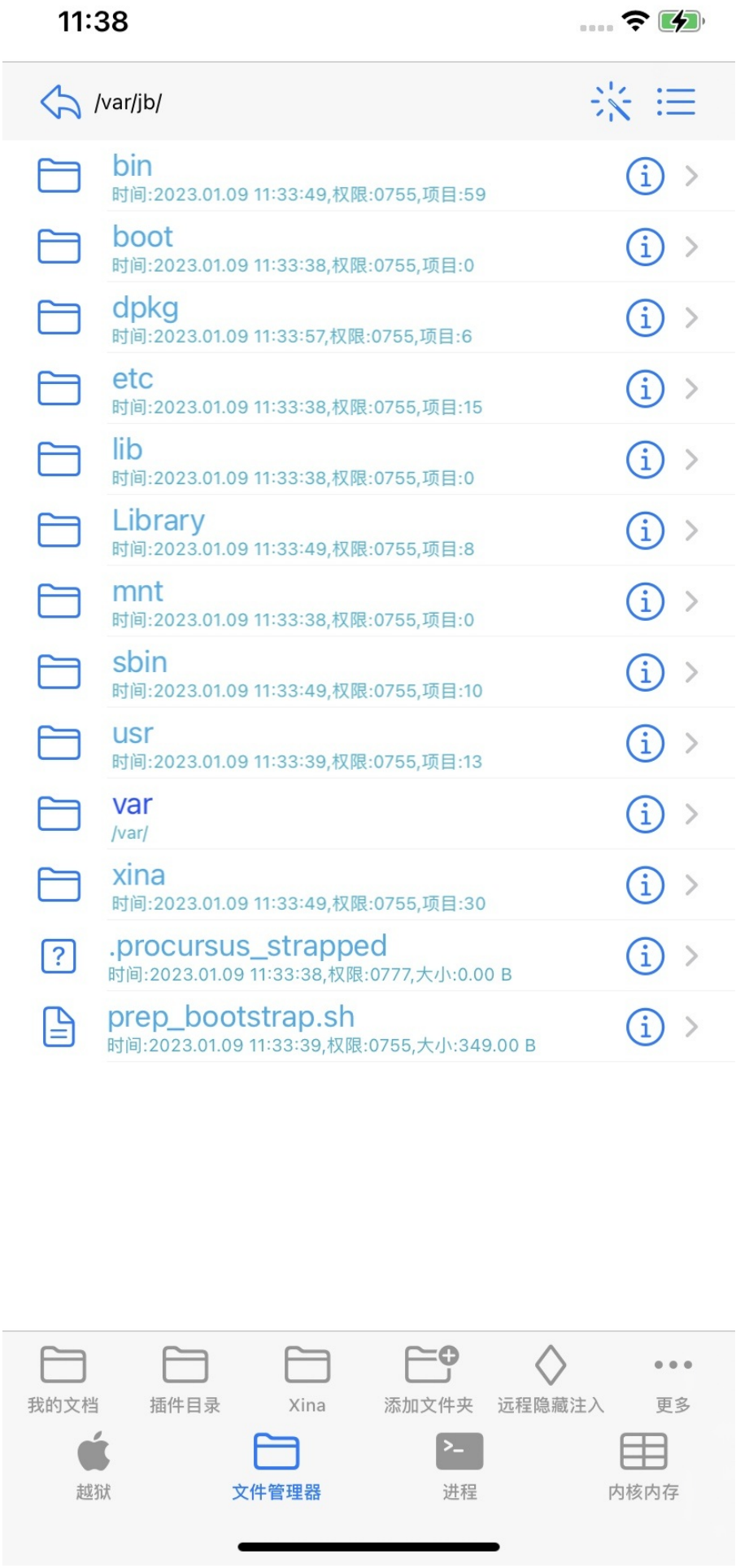
- 1.1.6.2
  - 主界面
    -


14:49
设置
注销
软重启
进入临退
退出
注意
Disclaimer: This version is a voluntary test version!...
该版本功劳属于你们
@LinusHenze @jaakerblom @zhuowei @tihmstar
@CStar_AD @Jakeashacks @opa334dev
语言 (Lang...
中文
EN
自动启动 ssh 服务
重新安装越狱环境
手动注入模式 (暂不使用)
安全模试 (出问题用)
当前版本: 1.1.6.2
开启越狱
卸载越狱
修复 dylib
越狱
文件管理器
进程
内核内存

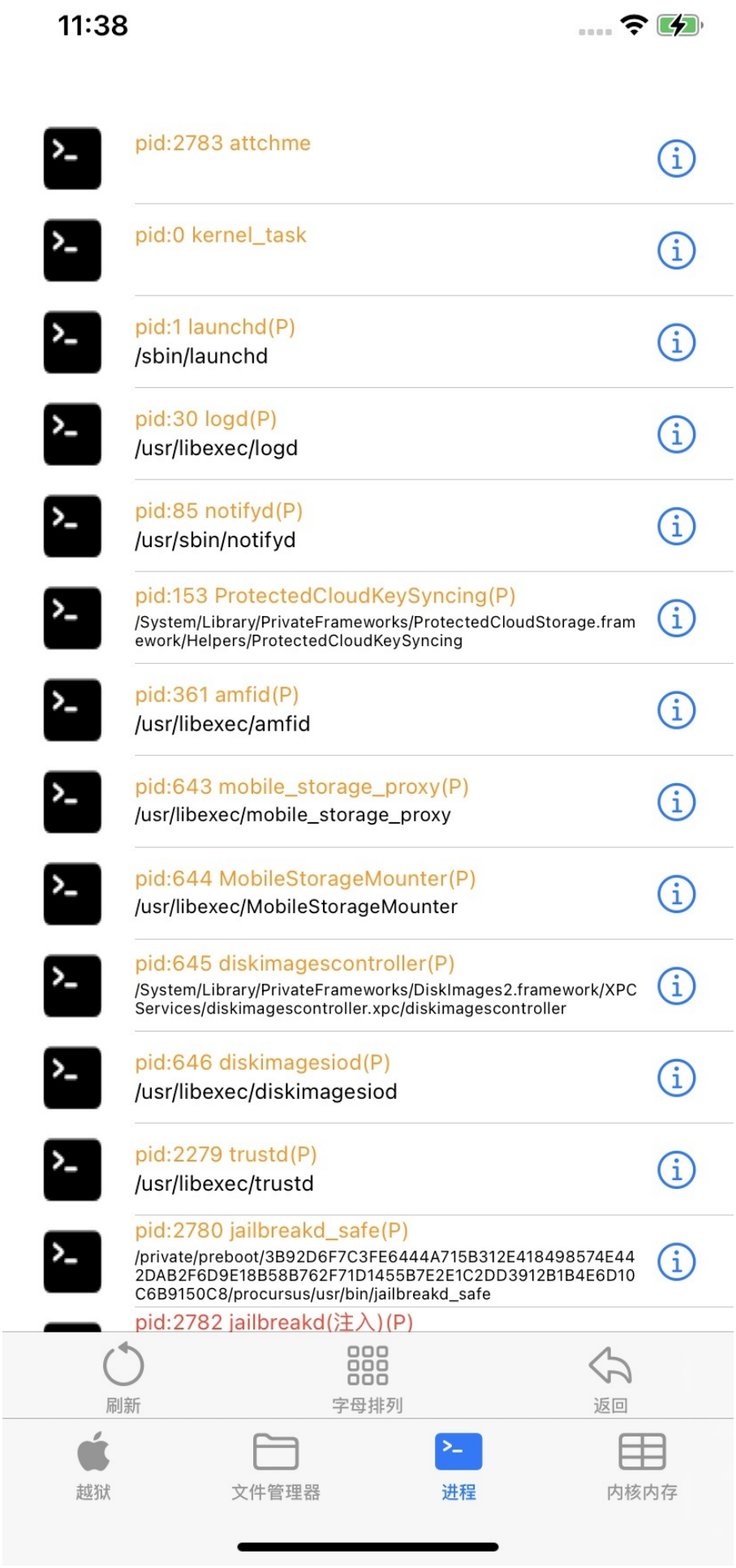
- 设置
  -


14:50
设置
注销
软重启
进入临退
退出
当前版本: 1.1.6.2
all_proc=ffffffff009ed8a60:
程序目录/private/var/containers/Bundle/Application/57EF4709-492D-4113-AA47-96B1E07B0B5D/XinaA12.app/XinaA12
程序文档目录/var/mobile/Containers/Data/Application/1ABF1C41-AEF4-4626-9B20-72D554F80CC6/Documents
当前已在越狱状态中.....
开启越狱
卸载越狱
修复 dylib
越狱
文件管理器
进程
内核内存

- 文件管理器
  -


11:38
/var/jb/
bin
时间:2023.01.09 11:33:49,权限:0755,项目:59
boot
时间:2023.01.09 11:33:38,权限:0755,项目:0
dpkg
时间:2023.01.09 11:33:57,权限:0755,项目:6
etc
时间:2023.01.09 11:33:38,权限:0755,项目:15
lib
时间:2023.01.09 11:33:38,权限:0755,项目:0
Library
时间:2023.01.09 11:33:49,权限:0755,项目:8
mnt
时间:2023.01.09 11:33:38,权限:0755,项目:0
sbin
时间:2023.01.09 11:33:49,权限:0755,项目:10
usr
时间:2023.01.09 11:33:39,权限:0755,项目:13
var
/var/
xina
时间:2023.01.09 11:33:49,权限:0755,项目:30
.procursus_strapped
时间:2023.01.09 11:33:38,权限:0777,大小:0.00 B
prep_bootstrap.sh
时间:2023.01.09 11:33:39,权限:0755,大小:349.00 B
我的文档
插件目录
Xina
添加文件夹
远程隐藏注入
更多
越狱
文件管理器
进程
内核内存

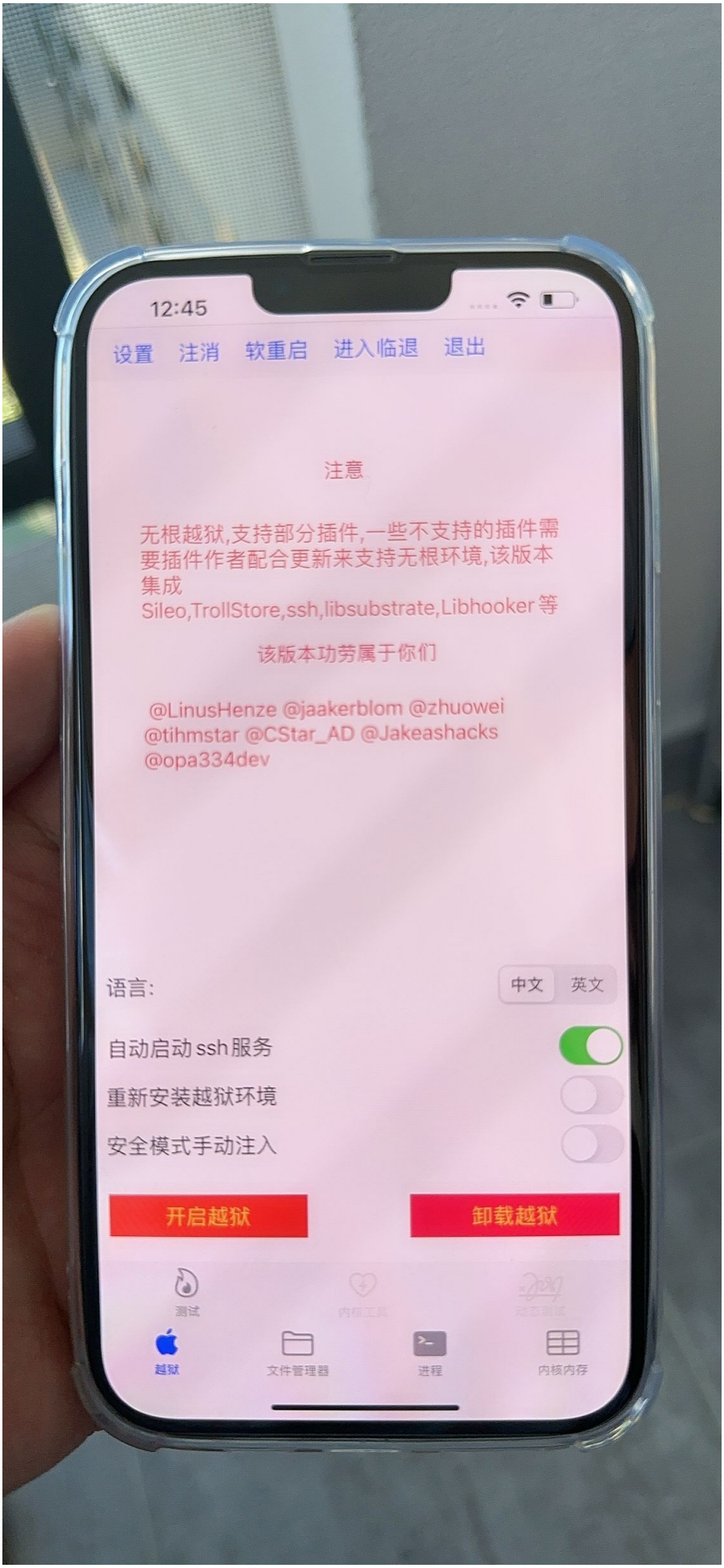
- 进程
  -


11:38
pid:2783 attchme
pid:0 kernel_task
pid:1 launchd(P)
/sbin/launchd
pid:30 logd(P)
/usr/libexec/logd
pid:85 notifyd(P)
/usr/sbin/notifyd
pid:153 ProtectedCloudKeySyncing(P)
/System/Library/PrivateFrameworks/ProtectedCloudStorage.framework/Helpers/ProtectedCloudKeySyncing
pid:361 amfid(P)
/usr/libexec/amfid
pid:643 mobile_storage_proxy(P)
/usr/libexec/mobile_storage_proxy
pid:644 MobileStorageMounter(P)
/usr/libexec/MobileStorageMounter
pid:645 diskimagescontroller(P)
/System/Library/PrivateFrameworks/DiskImages2.framework/XPCServices/diskimagescontroller.xpc/diskimagescontroller
pid:646 diskimagesiod(P)
/usr/libexec/diskimagesiod
pid:2279 trustd(P)
/usr/libexec/trustd
pid:2780 jailbreakd_safe(P)
/private/preboot/3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B762F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procursus/usr/bin/jailbreakd_safe
pid:2782 jailbreakd(注入)(P)
刷新
字母排列
返回
越狱
文件管理器
进程
内核内存

- 内核内存
  -


11:39
0xFFFFFF
内存查看
回退
越狱
文件管理器
进程
内核内存

- 更早版本的界面
  -


12:45
设置
注消
软重启
进入临退
退出
注意
无根越狱,支持部分插件,一些不支持的插件需要插件作者配合更新来支持无根环境,该版本集成
Sileo,TrollStore,ssh,libsubstrate,Libhooker等
该版本功劳属于你们
@LinusHenze @jaakerblom @zhuowei
@tihmstar @CStar_AD @Jakeashacks
@opa334dev
语言:
中文
英文
自动启动ssh服务
重新安装越狱环境
安全模式手动注入
开启越狱
卸载越狱
测试
越狱
文件管理器
进程
内核内存

XinaA15 1.1.8之后：新的UI界面

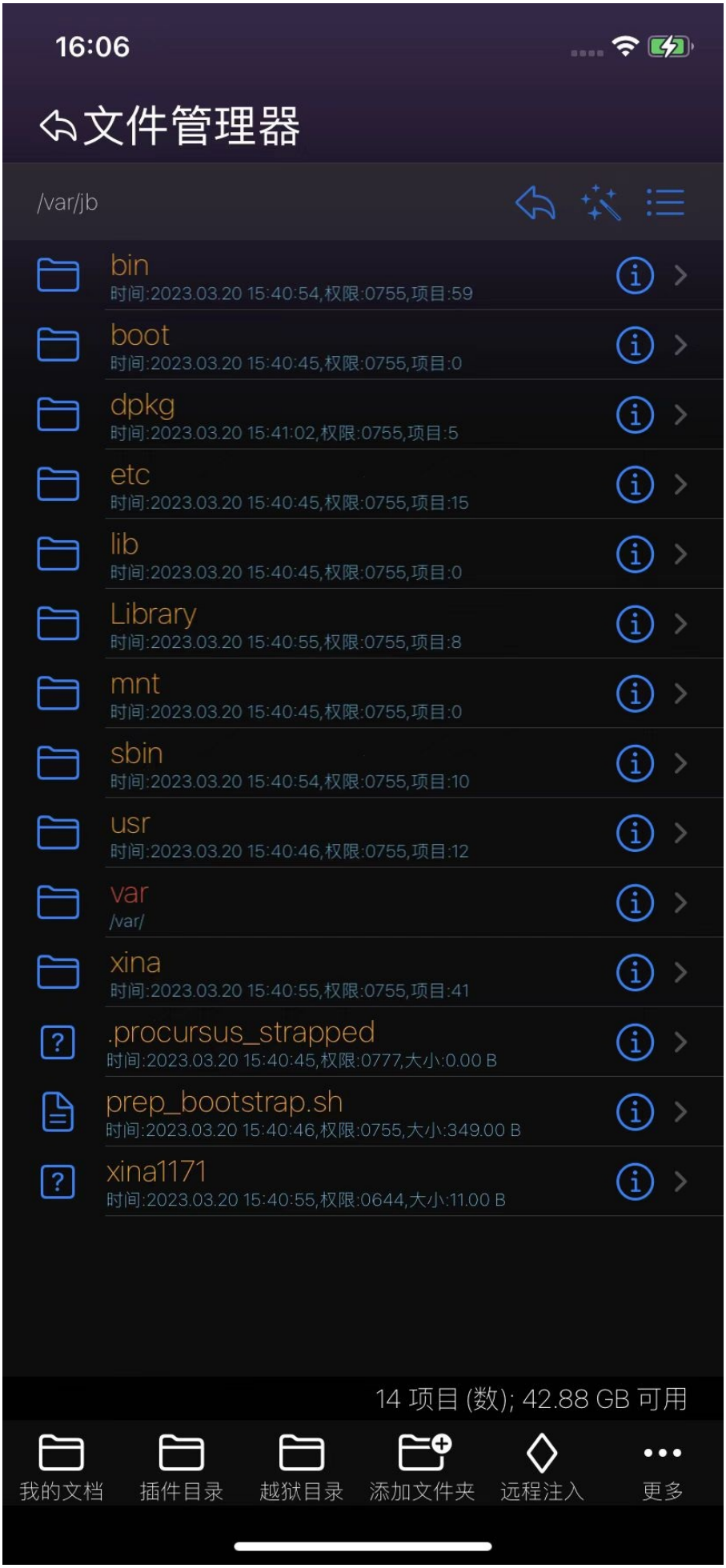
- 1.1.8
  - 主界面
    -


16:08
中文
EN
Xina iOS15 越狱工具
by Xina520
For iOS15.0 – iOS15.1.1
UI by Alierya
免责声明:使用该软件造成一切后果由自己承担!
get kernelbase=fffffff02aac4000;
get kernelslide=23ac0000;
get allproc=fffffff009ed8a60;
uid=0
Process directory=/private/var/containers/Bundle/Application/DF6A47F9-E725-4EDE-AA94-A00CB9EC256C/XinaA12.app/XinaA12
Program document directory=/var/mobile/Containers/Data/Application/1832D574-D4E9-4C89-98FC-72E187E37988/Library
当前已在越狱状态中.....
注销
软重启
临退
重新越狱
当前版本: 1.1.8

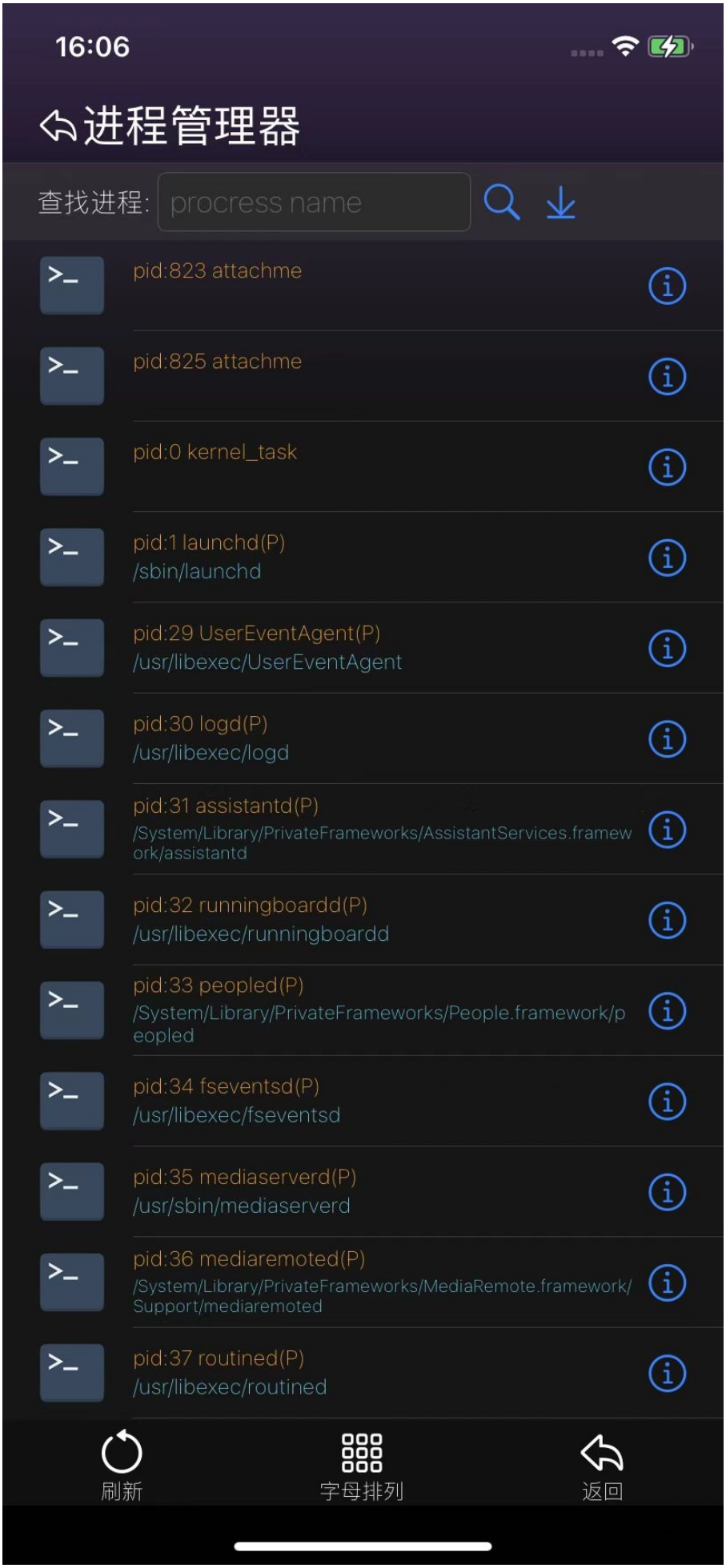
- 设置
  -


16:10
设置
部分功能请谨慎使用
文件管理
进程查看
内核查看
修复 dylib
自动启动 ssh 服务
重新安装越狱环境
安全模式 (有问题再开启)
首次自动安装巨魔
卸载程序
查看更新日志
更新下载连接
鸣谢:该工具功劳属于你们
@CStar_AD @Jakeashacks @opa334dev
@LinusHenze @jaakerblom @zhuowei @tihmstar

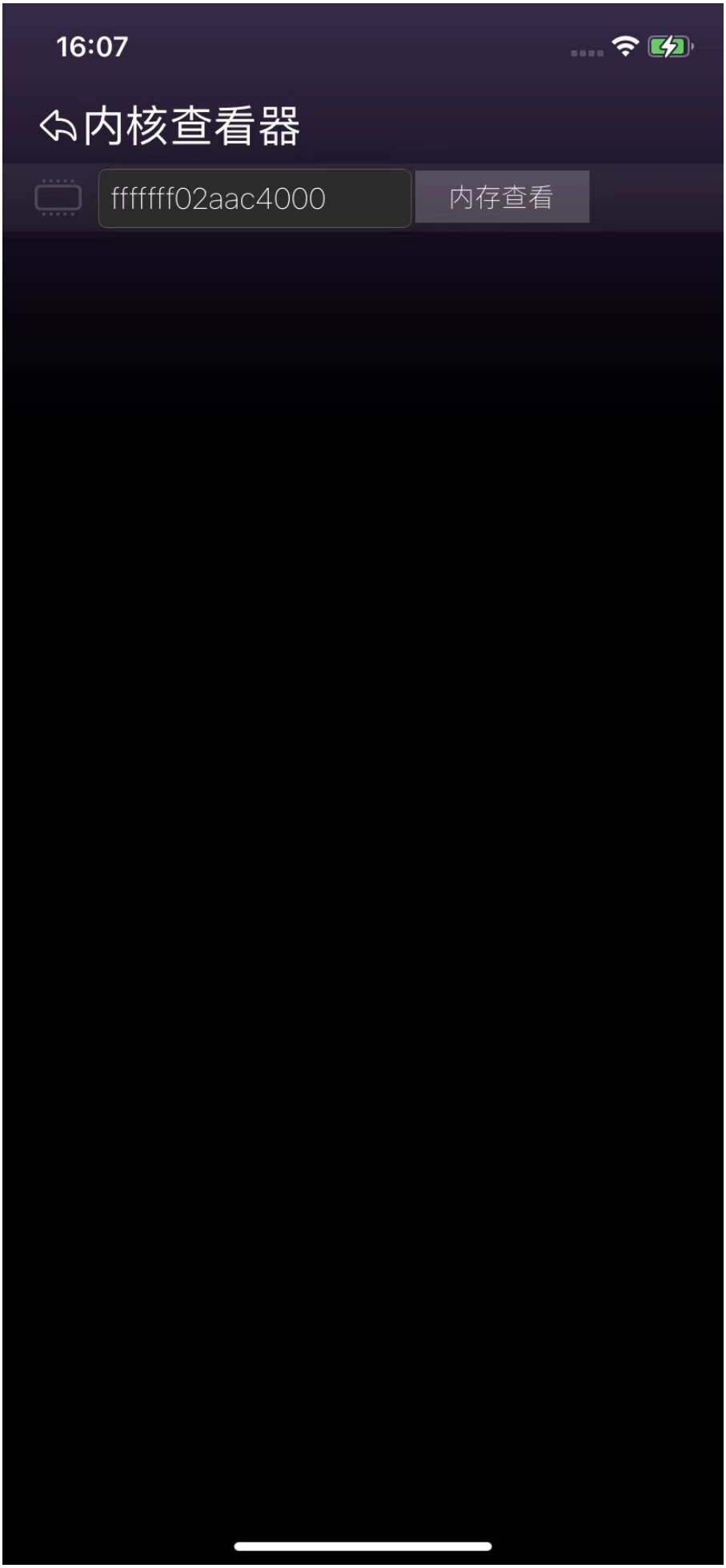
- 其他功能模块
  - 文件管理器
    -


16:06
文件管理器
/var/jb
bin
时间:2023.03.20 15:40:54,权限:0755,项目:59
boot
时间:2023.03.20 15:40:45,权限:0755,项目:0
dpkg
时间:2023.03.20 15:41:02,权限:0755,项目:5
etc
时间:2023.03.20 15:40:45,权限:0755,项目:15
lib
时间:2023.03.20 15:40:45,权限:0755,项目:0
Library
时间:2023.03.20 15:40:55,权限:0755,项目:8
mnt
时间:2023.03.20 15:40:45,权限:0755,项目:0
sbin
时间:2023.03.20 15:40:54,权限:0755,项目:10
usr
时间:2023.03.20 15:40:46,权限:0755,项目:12
var
/var/
xina
时间:2023.03.20 15:40:55,权限:0755,项目:41
.procursus_strapped
时间:2023.03.20 15:40:45,权限:0777,大小:0.00 B
prep_bootstrap.sh
时间:2023.03.20 15:40:46,权限:0755,大小:349.00 B
xina1171
时间:2023.03.20 15:40:55,权限:0644,大小:11.00 B
14 项目 (数); 42.88 GB 可用
我的文档
插件目录
越狱目录
添加文件夹
远程注入
更多

- 进程管理器
  -


16:06
进程管理器
查找进程:
procress name
pid:823 attachme
pid:825 attachme
pid:0 kernel_task
pid:1 launchd(P)
/sbin/launchd
pid:29 UserEventAgent(P)
/usr/libexec/UserEventAgent
pid:30 logd(P)
/usr/libexec/logd
pid:31 assistantd(P)
/System/Library/PrivateFrameworks/AssistantServices.framework/assistantd
pid:32 runningboardd(P)
/usr/libexec/runningboardd
pid:33 peopled(P)
/System/Library/PrivateFrameworks/People.framework/peopled
pid:34 fseventsd(P)
/usr/libexec/fseventsd
pid:35 mediaserverd(P)
/usr/sbin/mediaserverd
pid:36 mediaremoted(P)
/System/Library/PrivateFrameworks/MediaRemote.framework/Support/mediaremoted
pid:37 routined(P)
/usr/libexec/routined
刷新
字母排列
返回

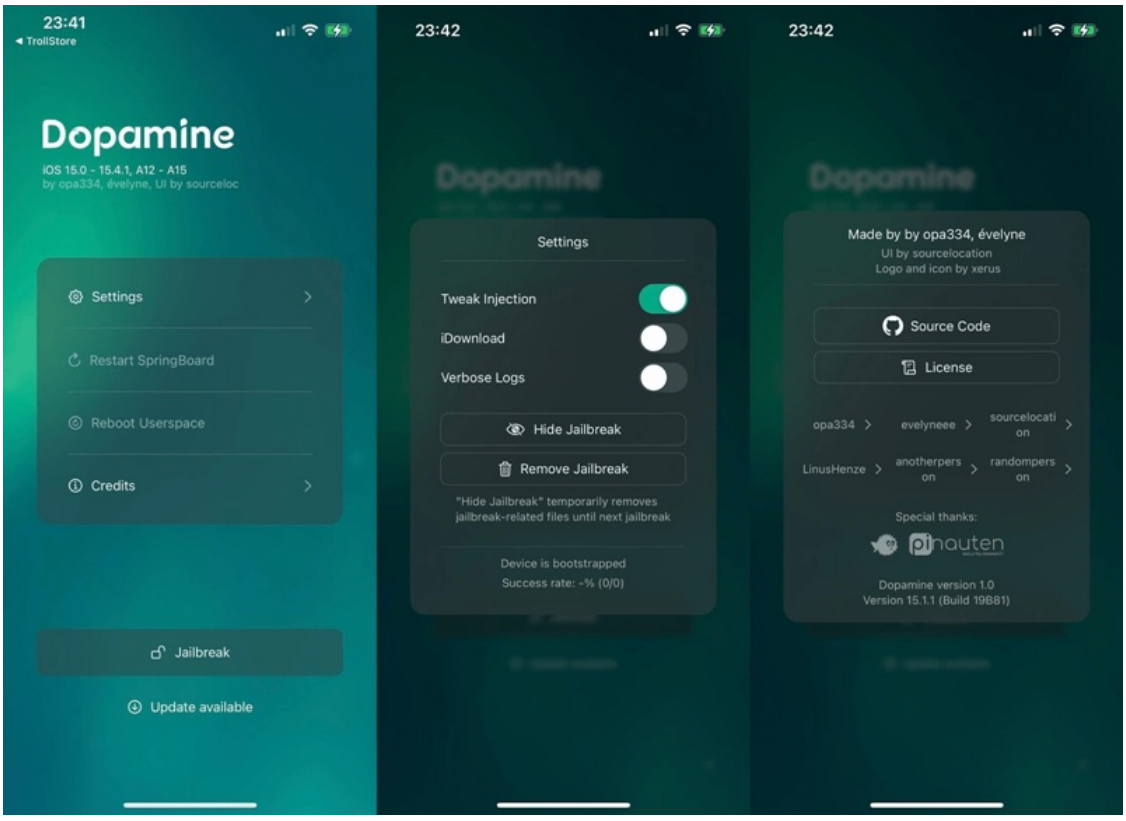
- 内核查看器
  -


16:07
内核查看器
ffffff02aac4000
内存查看

Dopamine

- Dopamine
  - 概述
    - iOS15的rootless越狱工具
      - a semi-untethered permasigned jailbreak for iOS 15
  - 越狱类型：`rootless jailbreak`
  - Logo
    - 
  - 名称
    - 旧称：`Fugu15 Max`
      - `Fugu15` 的作者：`Linus Henze`
  - 作者：`opa334` == `Lars Fröder`
    - Twitter：https://twitter.com/opa334dev
  - 资料
    - 官网
      - Dopamine Jailbreak (ellekit.space)
        - https://ellekit.space/dopamine/
    - Github
      - opa334/Dopamine: Dopamine is a semi-untethered permasigned jailbreak for iOS 15 (github.com)
        - https://github.com/opa334/Dopamine
  - 支持：
    - iOS/iPadOS版本：`15.0-15.4.1`
    - 芯片=CPU：
      - 具体型号：`A12+` == `A12` ~ `A15` 、M1
      - 对应的架构：`arm64e`
      - 对应的机型：iPhone XS, iPhone XS Max,iPhone XR, iPhone 11 Pro, iPhone 11 Pro Max, iPhone 11, iPhone 12 Pro, iPhone 12 Pro Max, iPhone 12, iPhone 12 mini, iPhone 13 Pro, iPhone 13 Pro Max, iPhone 13 mini, and iPhone 13
  - 基于：`Fugu15`
    - Fugu15
      - 功能=特点=机制
        - Automatic trust cache handling
        - 插件注入框架：`ElleKit`
          - `tweak injection framework` = 插件注入框架
          - = `tweak injection method` = 插件注入方法
          - = `tweak hook library` = 插件hook库
        - `libkrw` (including the ability to write to PPL protected memory and kcalling primitives)
      - 特殊：有个WiFi的bug
        - 说明：Dopamine已经修复此WiFi的bug了
  - 包管理器
    - 默认：`Sileo` 、`Zebra`
      - 当然可自行切换为别的，比如 `Saily`
  - 图

- 

- 

  - 其他
    - 越狱期间，需要临时关闭WiFi（越狱后，可正常开启Wifi）
    - 默认已加了rootless的软件源：Chariz、Havoc、Ellekit.space、Procurus、zp、BigBoss

附录

下面列出相关参考资料。

参考资料

- 【整理】iPhone机型和芯片型号和ARM架构
- 【未解决】给iOS 15.0的iPhone8用palera1n去越狱
- 【未解决】用palera1n的app去继续给iOS 15.0的iPhone8越狱
- 【已解决】palera1n越狱iPhone8出错：Whoops device did not enter DFU mode
- 【未解决】iOS 15越狱工具：palera1n
- 【整理】用palera1n给iOS 15.0的iPhone8越狱详细过程
- 【已解决】palera1n越狱出错：Could not download file The network connection was lost
- 【已解决】palera1n越狱的iPhone8中初始化ssh环境
- 【整理】越狱iPhone中常见解压缩插件工具下载地址
- 【已解决】Sileo中安装Filza报错：Depends zip unzip gzip unrar p7zip
- 【已解决】palera1n越狱后安装解压缩相关工具插件：尝试procurs的repo
- 【已解决】iPhone8重启后越狱丢失：给palera1n恢复越狱
- 【整理】iOS15越狱工具：XinaA15
- 【未解决】给iOS 15.1的iPhone11越狱
- 【已解决】用XinaA15给iOS 15.1的iPhone11越狱
- 【记录】iOS 15越狱工具：XinaA15
- 【已解决】购买iOS 15的二手iPhone
- 【记录】二手iOS 15的iPhone11到货了
- 【整理】iPhone中如何安装TrollStore
- 【已解决】iPhone中通过Safari安装TrollStore
- 【已解决】TrollStore安装ldid报错：Error downloading ldid 1001 请求超时
- 【已解决】给iOS 15.1的iPhone 11去翻墙科学上网安装代理
- 【未解决】iPhone中初始化配置TrollStore
- 【记录】越狱相关：TrollStore
- 【记录】TrollStore升级到最新版1.5.0
- 【已解决】iOS 15.1的iPhone11中安装TrollStore
- 【记录】重新用微信传输并打开再用TrollStore安装XinaA15最新版1.1.8的ipa
- 【已解决】卸载并重新用XinaA15给iPhone11越狱
- 【已解决】iOS 15.1的iPhone11中安装XinaA15
- 【已解决】如何安装XinaA15
- 【记录】下载XinaA15的ipa文件
- 【未解决】XinaA15越狱iPhone11后如何使用
- 【记录】用爱思助手查看XinaA15越狱后的iPhone信息
- 【已解决】XinaA15越狱后iPhone11中如何使用ssh
- 【记录】XinaA15越狱的iPhone11重启后越狱是否丢失
- 【记录】重启iPhone并恢复XinaA15的越狱
- 【已解决】XinaA15重新恢复越狱环境
- 【已解决】升级XinaA15到新版本1.1.8再重新越狱
- 【已解决】用新版1.1.8的XinaA15重新越狱
- 【记录】iPhone11重新安装XinaA15和重新用XinaA15越狱
- 【已确认】iPhone11中重新确认新版XinaA和新版Frida是否可以正常使用
- 【记录】看看新版1.1.8的XinaA15各个功能和界面
- 【已解决】iOS逆向：卸载XinaA15
- 【记录】iPhone11升级1.1.8新版XinaA15重新越狱后
- 【记录】升级XinaA15后重新初始化TrollStore巨魔
- 【未解决】iOS 15的iPhone越狱
- 【整理】iOS15越狱相关：越狱插件软件源
- 【整理】UICache是什么和作用
- 【整理】新的iOS15越狱工具：Dopamine

- 【已解决】从XinaA15越狱后的iPhone11中找支持arm64e的libsubstrate.dylib
- 【已解决】iOSOpenDev的插件dylib注入iPhone11失败：mach-o file but is an incompatible architecture have arm64 need arm64e
-