
目录

前言	1.1
iOS15越狱概览	1.2
iOS15+越狱背景知识	1.3
rootless和rootful	1.3.1
iPhone机型信息	1.3.2
免签安装ipa文件	1.3.3
palera1n	1.4
越狱前	1.4.1
前提条件	1.4.1.1
注意事项和说明	1.4.1.2
工具的版本	1.4.1.3
文档和资料	1.4.1.4
越狱中	1.4.2
palera1n的rootful越狱概述	1.4.2.1
palera1n的rootful越狱详解	1.4.2.2
常见问题	1.4.2.3
help语法	1.4.2.4
loader app	1.4.2.5
越狱后	1.4.3
ssh	1.4.3.1
如何恢复越狱	1.4.3.2
常见问题	1.4.3.3
XinaA15	1.5
越狱前	1.5.1
历史版本和下载地址	1.5.1.1
越狱中	1.5.2
安装XinaA15	1.5.2.1
用XinaA15越狱	1.5.2.2
越狱后	1.5.3
界面和功能	1.5.3.1
Dopamine	1.6
附录	1.7
参考资料	1.7.1

iOS逆向：iOS15越狱

- 最新版本： `v1.2.1`
- 更新时间： `20240318`

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；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

此电子书教程的全部内容，如无特别说明，均为本人原创。其中部分内容参考自网络，均已备注了出处。如发现有侵权，请通过邮箱联系我 `admin` 艾特 `crifan.com`，我会尽快删除。谢谢合作。

各种技术类教程，仅作为学习和研究使用。请勿用于任何非法用途。如有非法用途，均与本人无关。

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](#)

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2024-03-19 09:16:33

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](#)

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2023-06-28 22:08:57

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。

crifan.org，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新：2023-06-28 22:11:01

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

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2023-06-28 22:10:43

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/procurus/xina/libsubstrate.dylib
 - 大小： 51KB
 - 只支持一种架构： arm64e
 - /private/preboot/3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B762F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procurus/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/procurus/Library/MobileSubstrate/DynamicLibraries/jailAppleAccount.dylib
默认 14:50:37.206560+0800 jailbreakd /private/preboot/3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B762F71D1455B7E2E1C2DD3912B1B4E6D10C6B9150C8/procurus/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/procurus/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
```

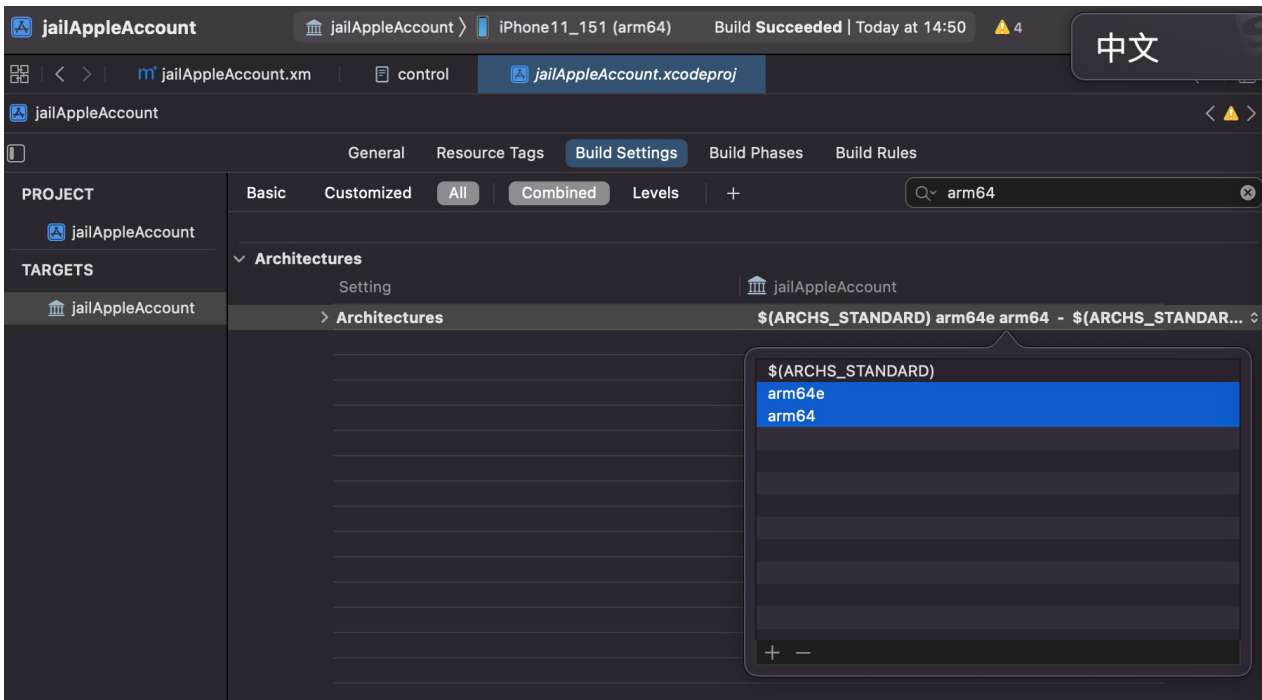
```
reboot/3B92D6F7C3FE6444A715B312E418498574E442DAB2F6D9E18B58B762F71D1455B7E2E1C2DD3912B1B4E0D10C6B9150C8/procursus/Library/MobileSubstrate/DynamicLibraries/jailAppleAccount.dylib' (mach-o file, but is an incompatible architecture (have ...
```

- 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



crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2024-03-18 09:56:43

免签安装ipa文件

免签安装ipa文件，可以用：

- [TrollStore](#)
- [Sideloadly](#)

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新：2024-03-18 09:50:39

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

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2023-06-28 22:37:43

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

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2023-06-28 22:27:05

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

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2023-06-28 22:27:59

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 = rootFu1`
- `palera1n == palera1n -l : l = rootLess`
- 完整用法=语法=帮助
 - `palera1n --help`
 - 或 在线的html文档
 - <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>

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2023-06-28 22:28:44

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-palera1x.md at main · palera1n/docs \(github.com\)](#)
- [docs/booting-palera1x.md at main · palera1n/docs \(github.com\)](#)
- [docs/jailbreak-with-palera1x.md at main · palera1n/docs \(github.com\)](#)

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2023-06-28 22:29:53

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

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2023-06-28 23:30:50

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`

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2023-06-28 22:36:03

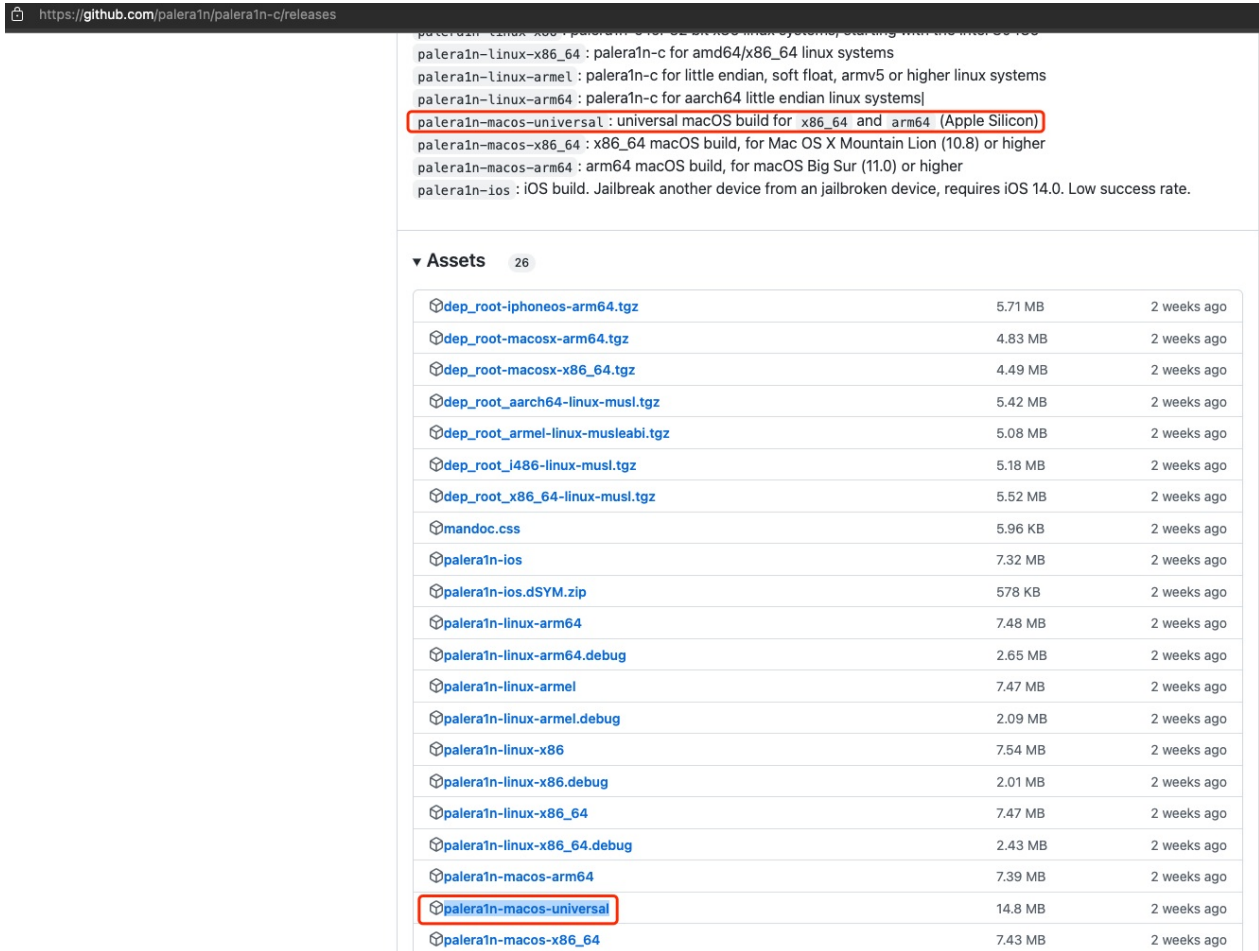
用palera1n给iOS 15.0的iPhone8越狱的详细过程

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>



The screenshot shows the GitHub release page for palera1n-c. The release version is v2.0.0-beta.4. The description lists various binaries for different architectures and operating systems. The file 'palera1n-macos-universal' is highlighted with a red box, indicating it is the target for Mac users. Below the description, the 'Assets' section lists 26 files, with 'palera1n-macos-universal' also highlighted in red.

Asset Name	Size	Uploaded
dep_root-iphoneos-arm64.tgz	5.71 MB	2 weeks ago
dep_root-macosx-arm64.tgz	4.83 MB	2 weeks ago
dep_root-macosx-x86_64.tgz	4.49 MB	2 weeks ago
dep_root_aarch64-linux-musl.tgz	5.42 MB	2 weeks ago
dep_root_armel-linux-musleabi.tgz	5.08 MB	2 weeks ago
dep_root_i486-linux-musl.tgz	5.18 MB	2 weeks ago
dep_root_x86_64-linux-musl.tgz	5.52 MB	2 weeks ago
mandoc.css	5.96 KB	2 weeks ago
palera1n-ios	7.32 MB	2 weeks ago
palera1n-ios.dSYM.zip	578 KB	2 weeks ago
palera1n-linux-arm64	7.48 MB	2 weeks ago
palera1n-linux-arm64.debug	2.65 MB	2 weeks ago
palera1n-linux-armel	7.47 MB	2 weeks ago
palera1n-linux-armel.debug	2.09 MB	2 weeks ago
palera1n-linux-x86	7.54 MB	2 weeks ago
palera1n-linux-x86.debug	2.01 MB	2 weeks ago
palera1n-linux-x86_64	7.47 MB	2 weeks ago
palera1n-linux-x86_64.debug	2.43 MB	2 weeks ago
palera1n-macos-arm64	7.39 MB	2 weeks ago
palera1n-macos-universal	14.8 MB	2 weeks ago
palera1n-macos-x86_64	7.43 MB	2 weeks ago

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), checkrain 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), checkrain 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 abdc0dd061c3cb96f5c4afe109de4eb48b88433a to enter recovery mode immediately
- [02/27/23 14:13:20] Info : Press Enter when ready for DFU mode
```

```

crifan@licrifandeMacBook-Pro ~ (-zsh)
65% 10 GB 8.2 kB↓ 39 kB↑ 2/27, 14:13
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.), Proculus Team (Hayden Seay, Cameron Katri, Keto et.al)

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)
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.), Proculus Team (Hayden Seay, Cameron Katri, Keto et.al)

- [ ] <Info>: Waiting for devices
- [ ] <Info>: Telling device with udid abdc0dd961c3cb96f5c4afe109de4eb48b88433a to enter recovery mode immediately
- [ ] <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：



然后去：

- Enter=回车

确认准备好，提示：`get ready`

```
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), checkrain team (Siguza, axi0mx, littlelailo
# et al.), Procursus Team (Hayden Seay, Cameron Katri, Keto et.al)
- [          ] <Info>: Waiting for devices
- [          ] <Info>: Telling device with udid abdc0dd961c3cb96f5c4afe109de4eb48b88433a to enter recovery mode immediately
- [          ] <Info>: Press Enter when ready for DFU mode
Get ready (2)
```

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 checkrain
#
# Checkrain 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, qwertyoruio, sbingner, siguza
#===== Thanks to =====
# haifisch, jndok, jonseals, xerub, lilstevie, psychotea, sferrini
# Cellaebrite (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:



直到看到最后的log: `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：

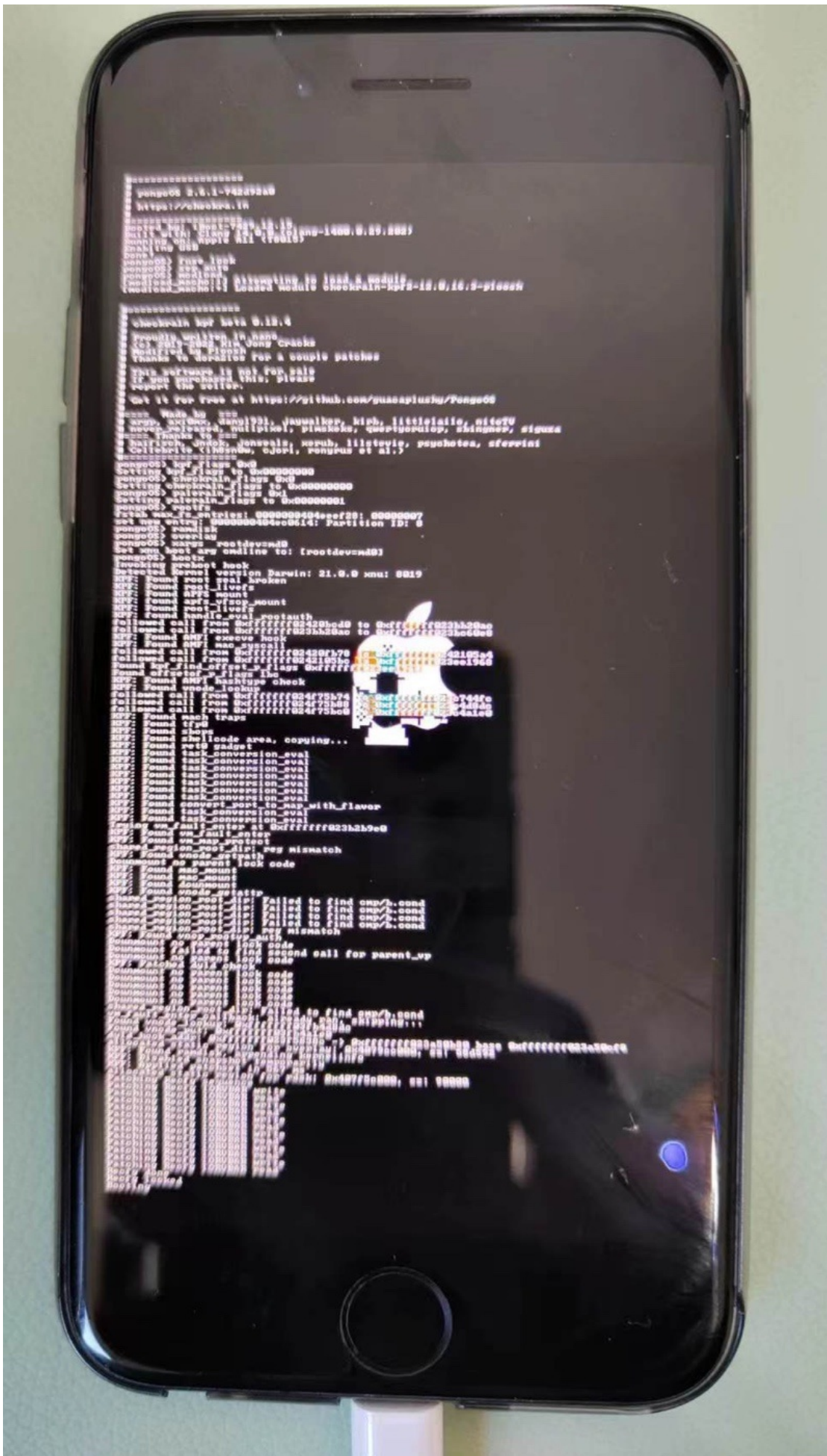
```
crifan@licrifandemacbook-pro ~/dev/dev_tool/reverse_security/iOS/palera1n$ palera1n -f
# == palera1n-c ==
#
# Made by: Nick Chan, Ploosh, Mineek, Nebula, llsc12
#
# Thanks to: dora2ios, pythonplayer, tihmstar, nikias
# (libimobiledevice), checkrain team (Siguza, axi0mx, littlelailo
# et al.), Procurus 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 abc0dd961c3cb96f5c4afe109de4eb48b88433a to enter recovery mode 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 checkrain
#
# Checkrain 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

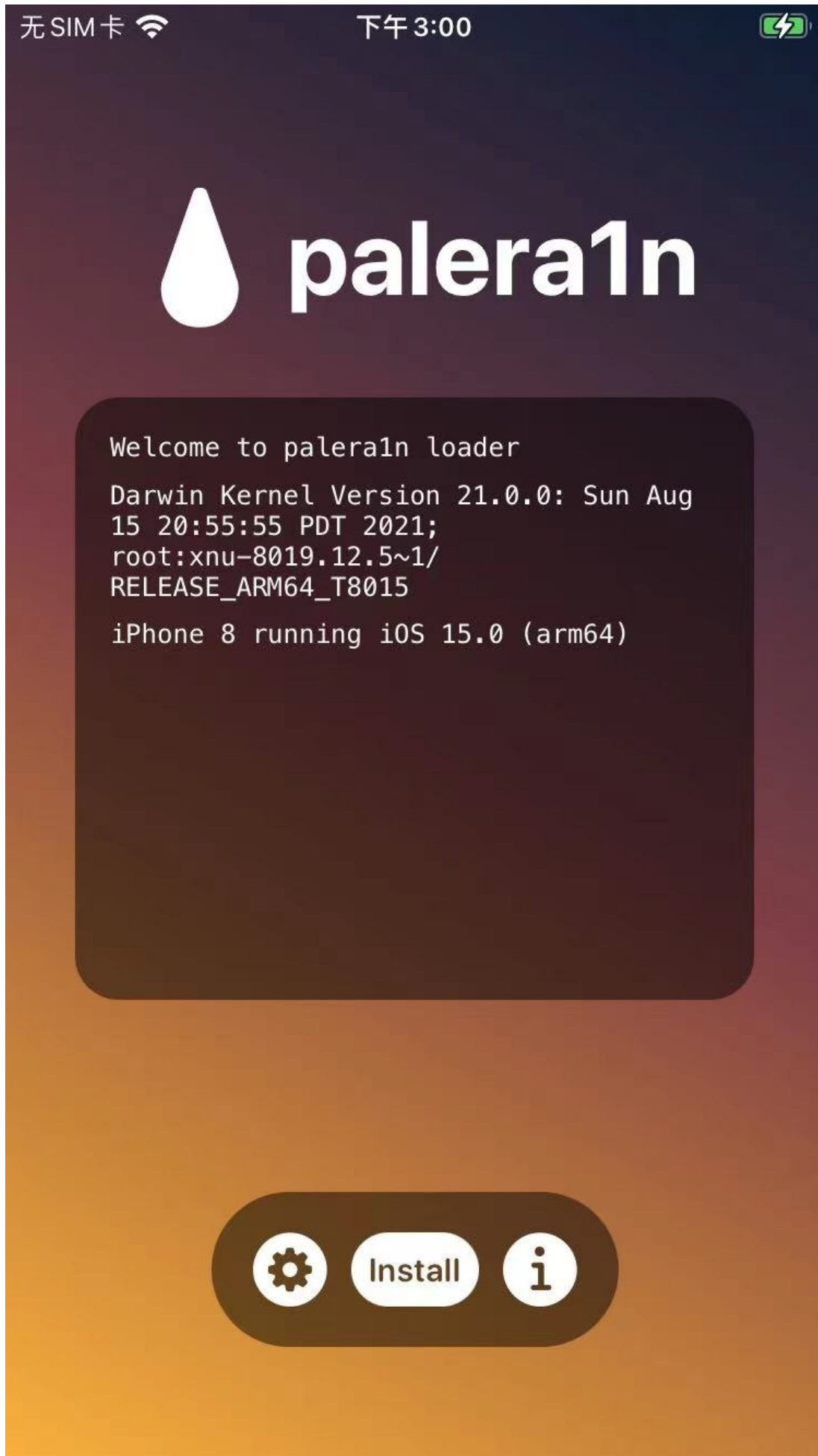
进入iPhone桌面后，可以看到：

palera1n的app = palera1n loader = palera1n的图标



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



此处会显示：

- 当前iPhone信息
 - iPhone 8 running iOS 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



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。

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2023-06-28 23:31:23

常见问题

Warning Whoops device did not enter DFU mode

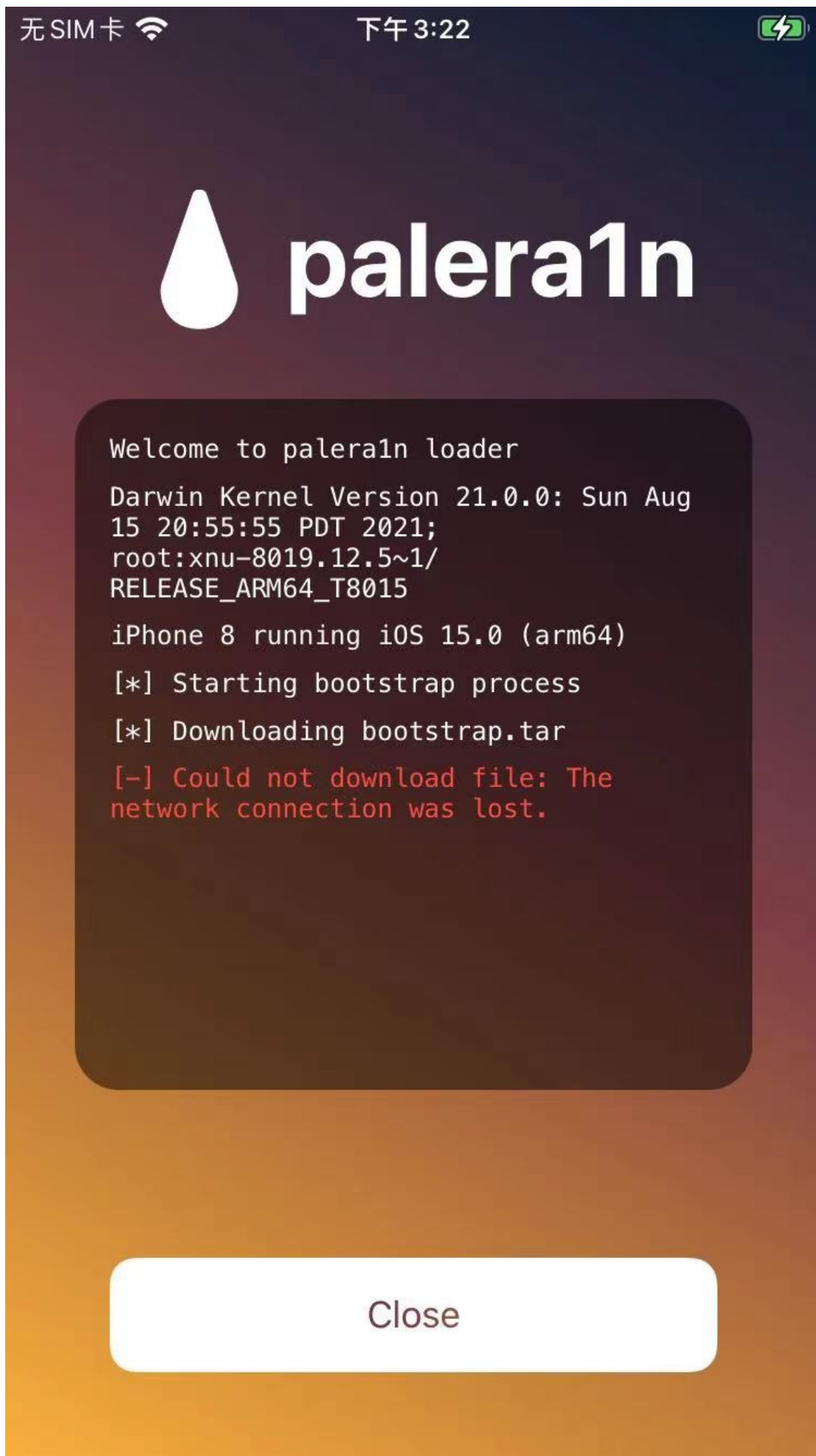
现象：尝试进入DFU模式时遇到：`<warning>: Whoops, device did not enter DFU mode`

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：



- 解决办法
 - 多试几次

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2023-06-28 23:35:41

palera1n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 directory (path the built-in checkra1n will be extracted to)

```

palera1n -l

用palera1n查看当前连接的iPhone的信息：

```

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

```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2023-06-28 22:32:01

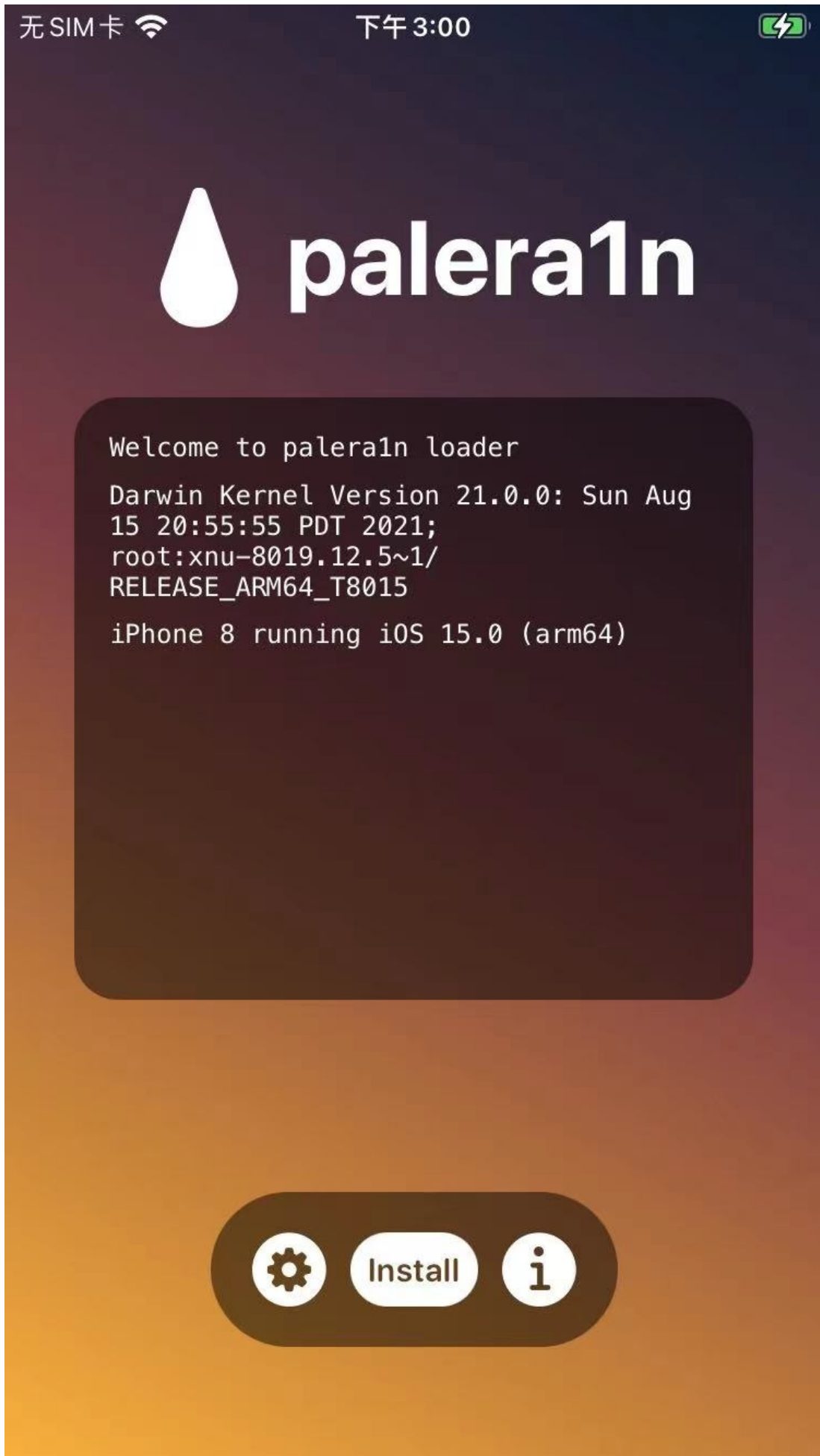
palera1n loader app

- palera1n的app = palera1n loader = palera1n loader app

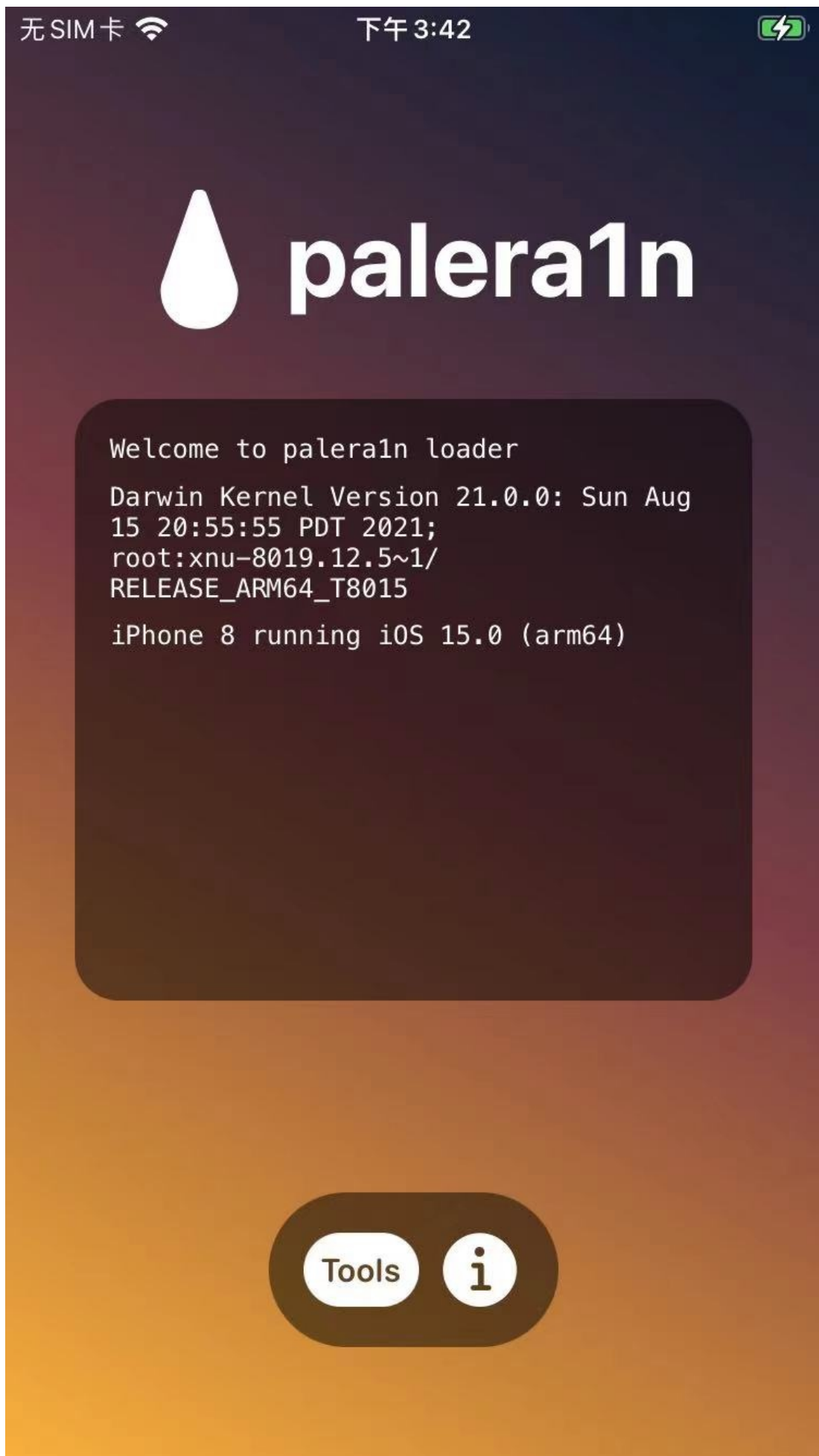
打开palera1n的app后:

首页

- 如果中间出现Install按钮: 说明越狱过程还没结束。
 -



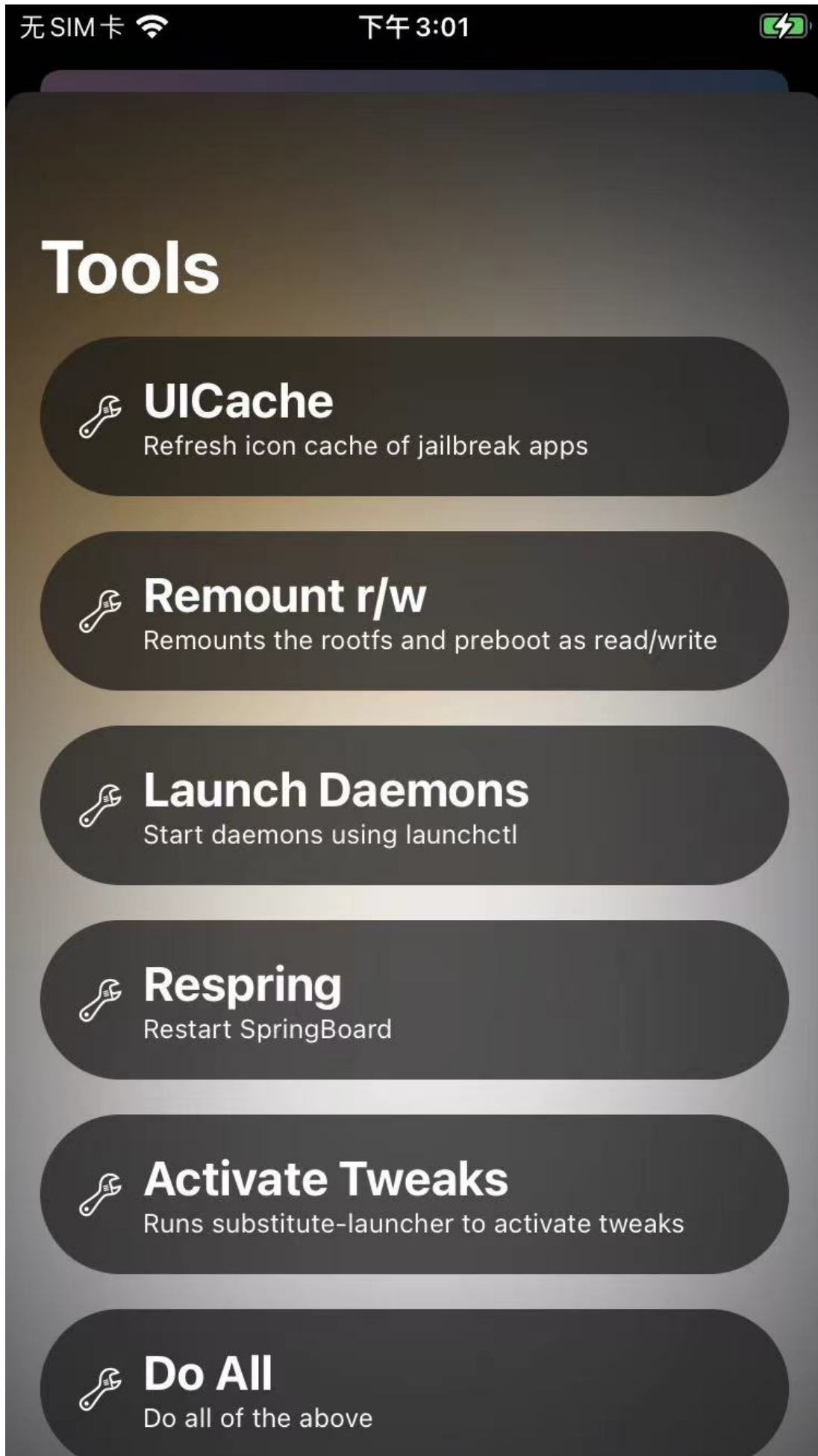
- 如果中间没有了Install按钮：说明已经越狱成功
 -

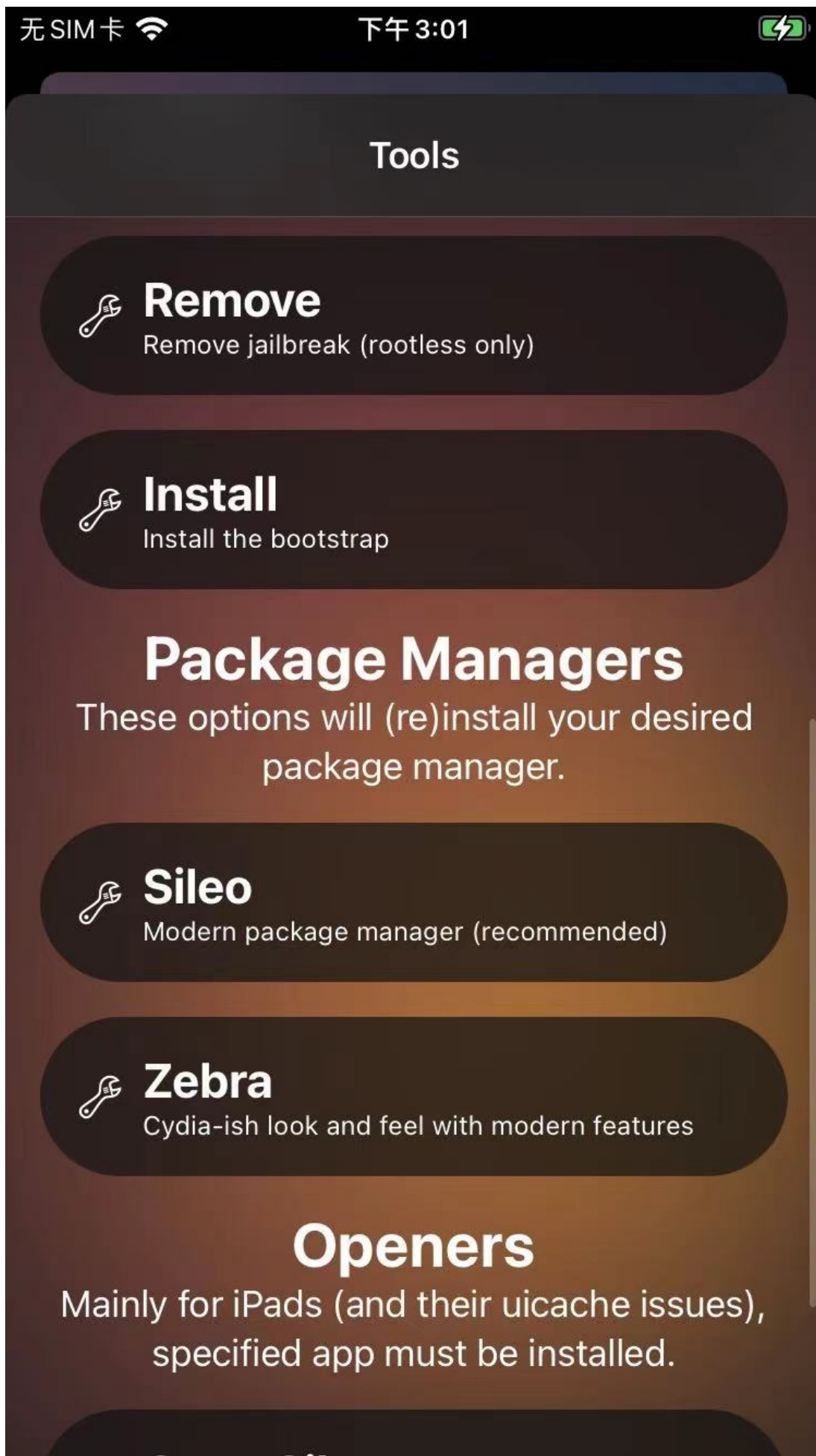


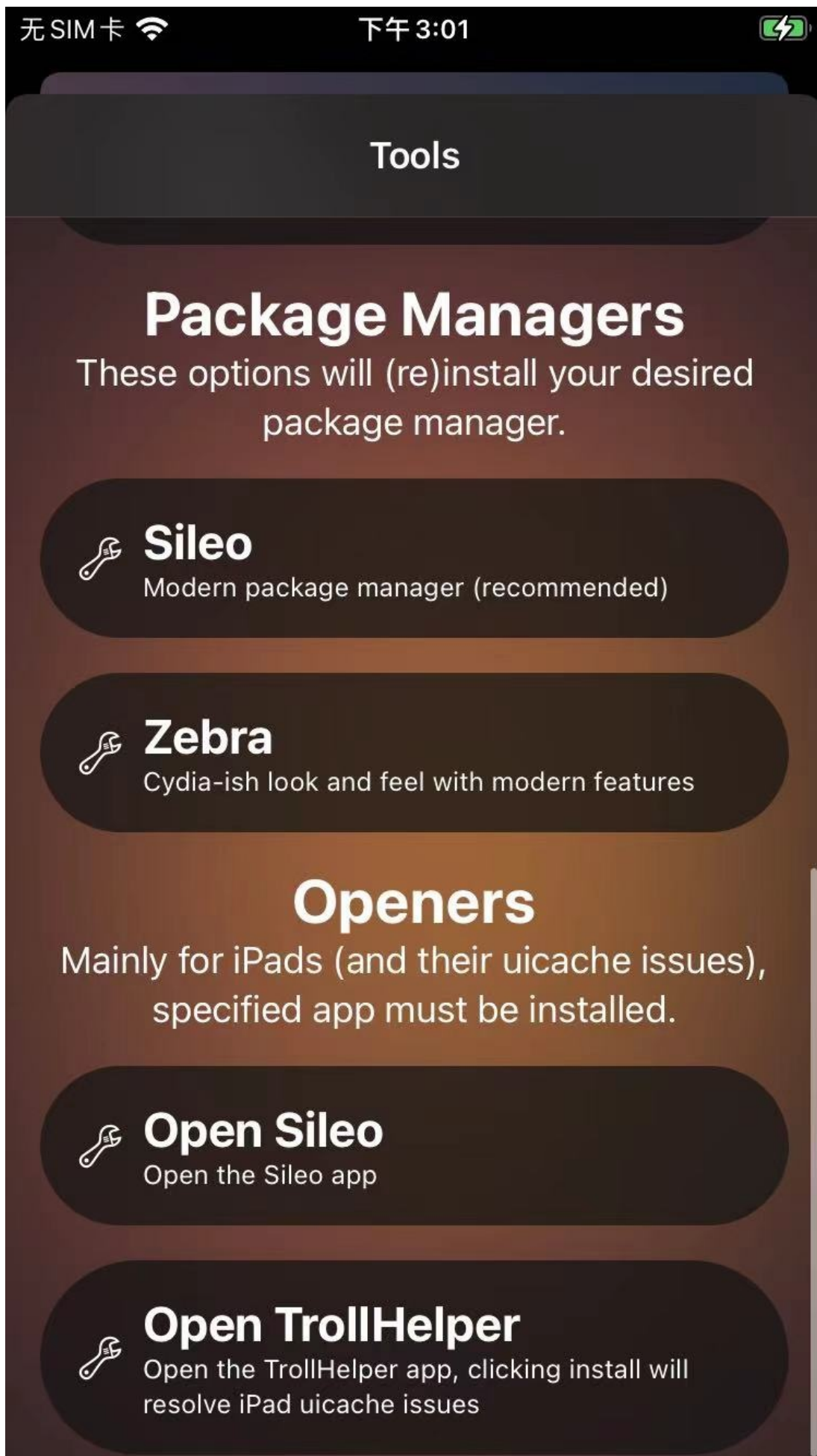
去看看palera1n的app的其他信息：

Tools

点击左边的齿轮图标=设置=工具=Tools，会打开Tools页面：

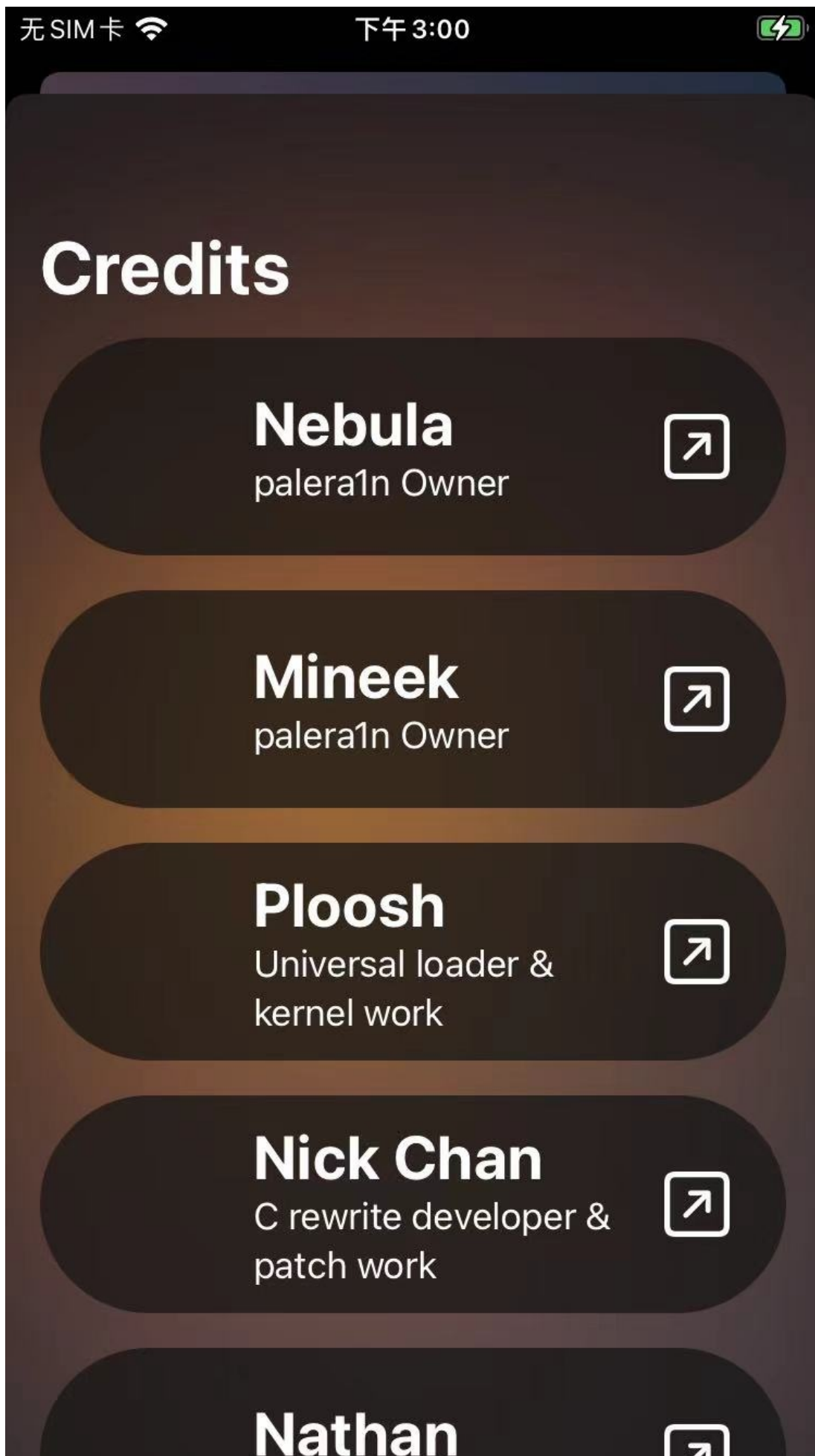


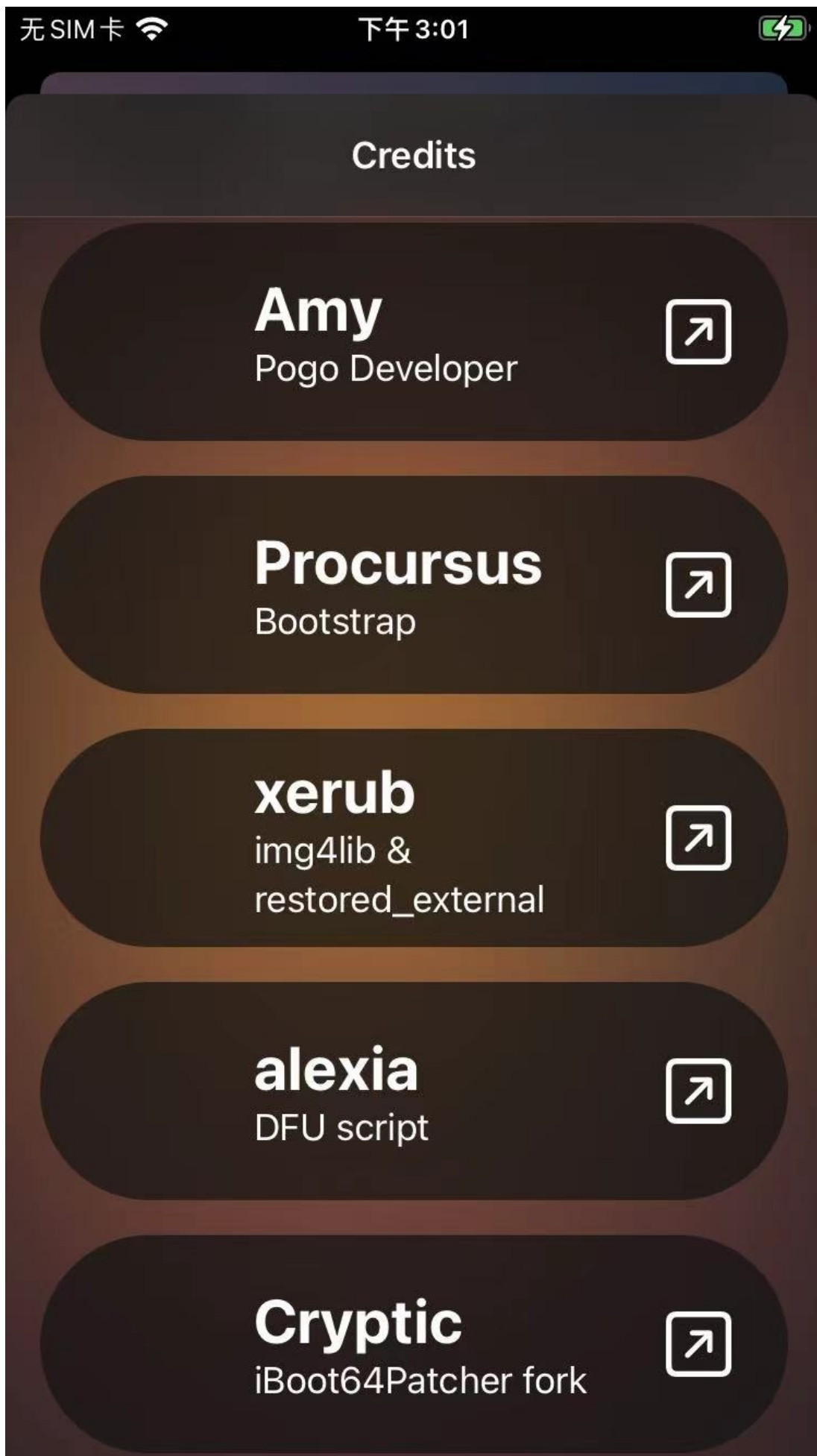




Credits

点击右边的i=info, 会打开Credits页面:





palera1n越狱后

能看到palera1n的进程

Mac中通过frida-ps查看的：

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

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新： 2023-06-28 22:40:01

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。

crifan.org，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新： 2023-06-28 22:39:17

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

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新： 2023-06-28 22:42:21

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://tigissoftware.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>

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新： 2023-06-28 22:41:19

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

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2023-06-28 22:43:36

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/i0FDG0qymm4b>
- 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>

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新：2023-06-28 22:44:16

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
 -





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。

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2023-06-28 23:37:47

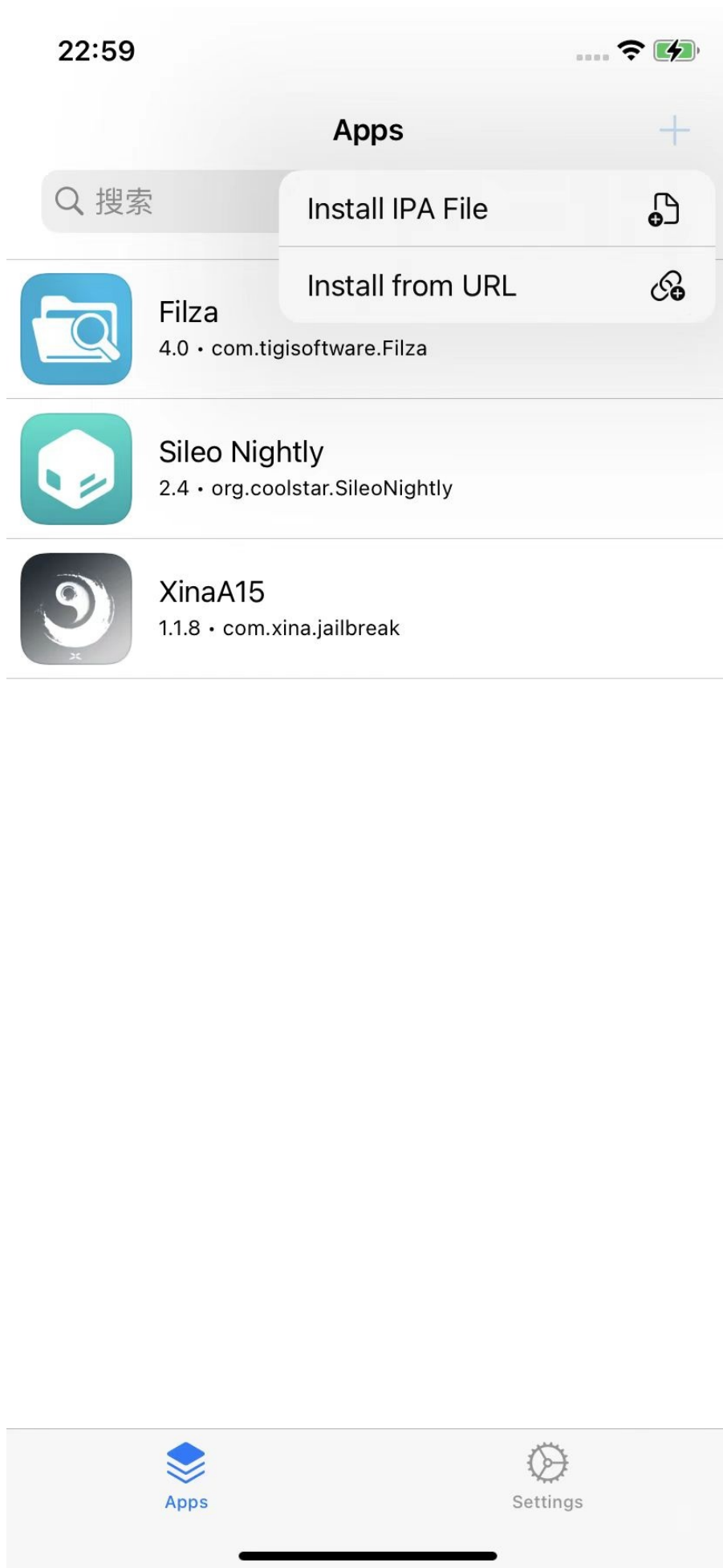
安装XinaA15

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
 -



- 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

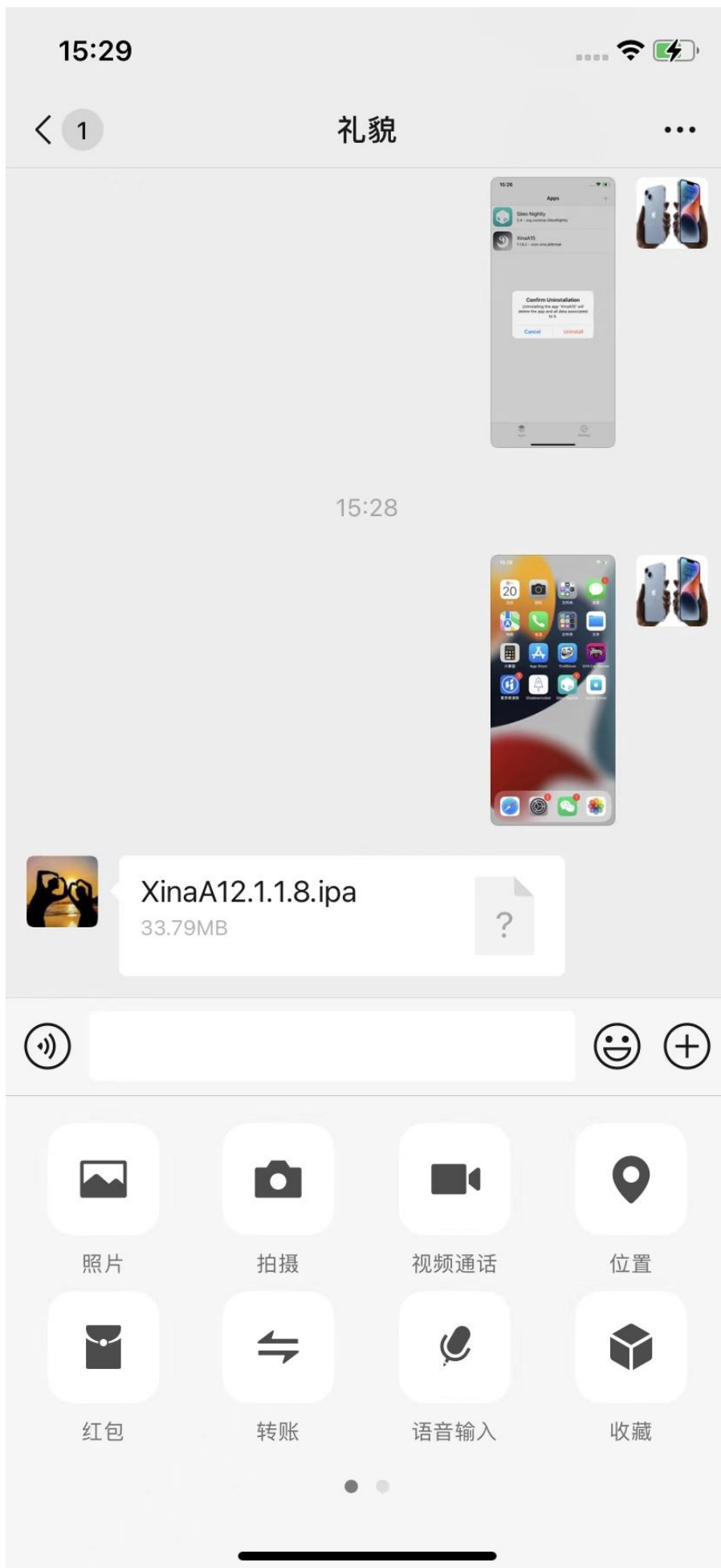
- 下载XinaA的ipa文件
 - 从官网 <https://xina.ss03.cn/> 下载到最新版本的XinaA15的ipa文件
 - 比如：1.1.8 的 <http://apt.xina.vip/XinaA12.1.1.8.ipa>
 - 注意：此处直接下载没速度，最后改用迅雷，可以顺利下载

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
 -



- 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

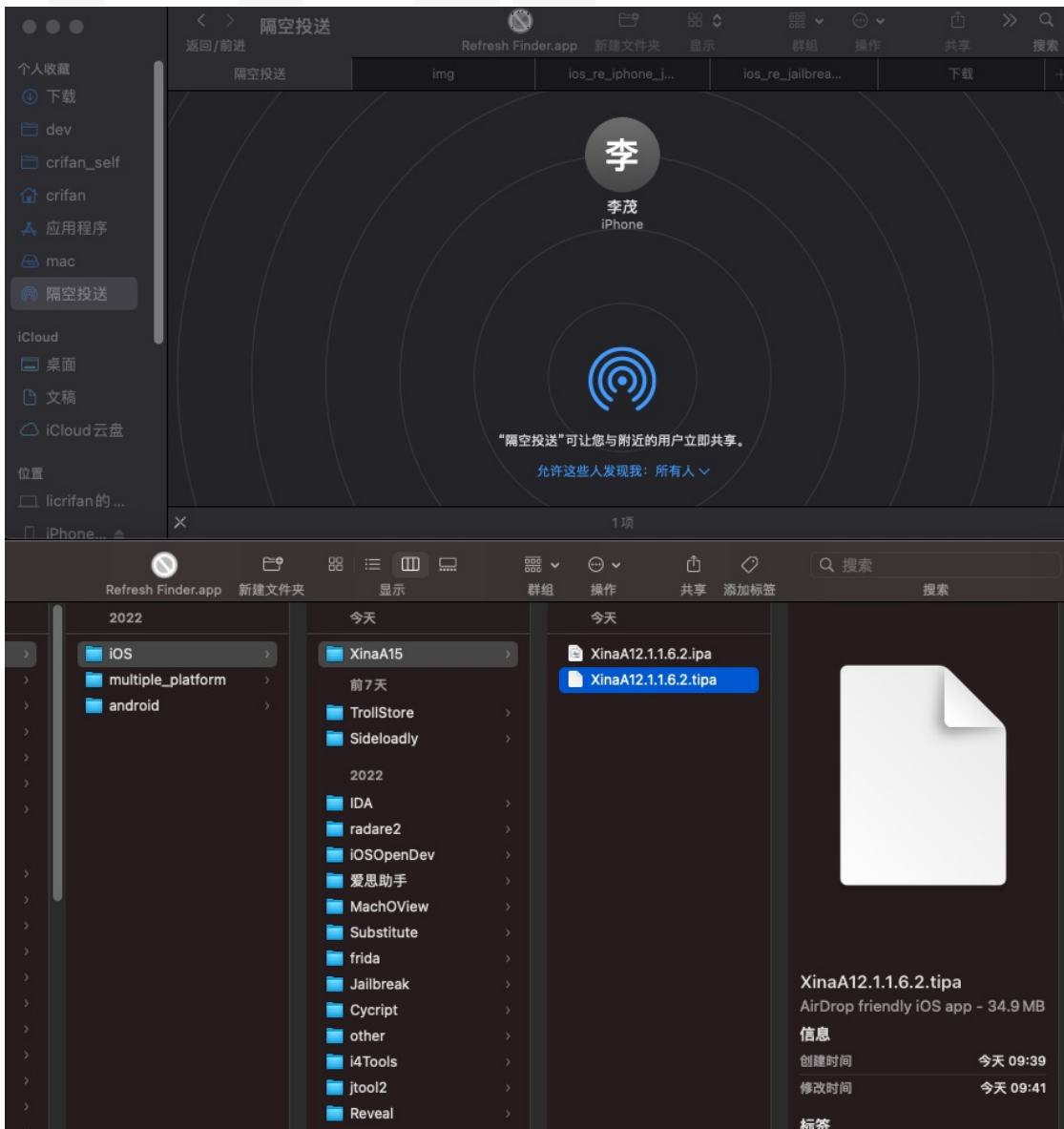
- 点击: TrollStore





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



- 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

25.2 MB



2月 06, 2023 17:07



XinaA12.1.1.6.2.ipa

34.9 MB



1月 09, 2023 09:41

[/dev/disk0s1s2](#) [42.4 GB](#) [2 item\(s\)](#)

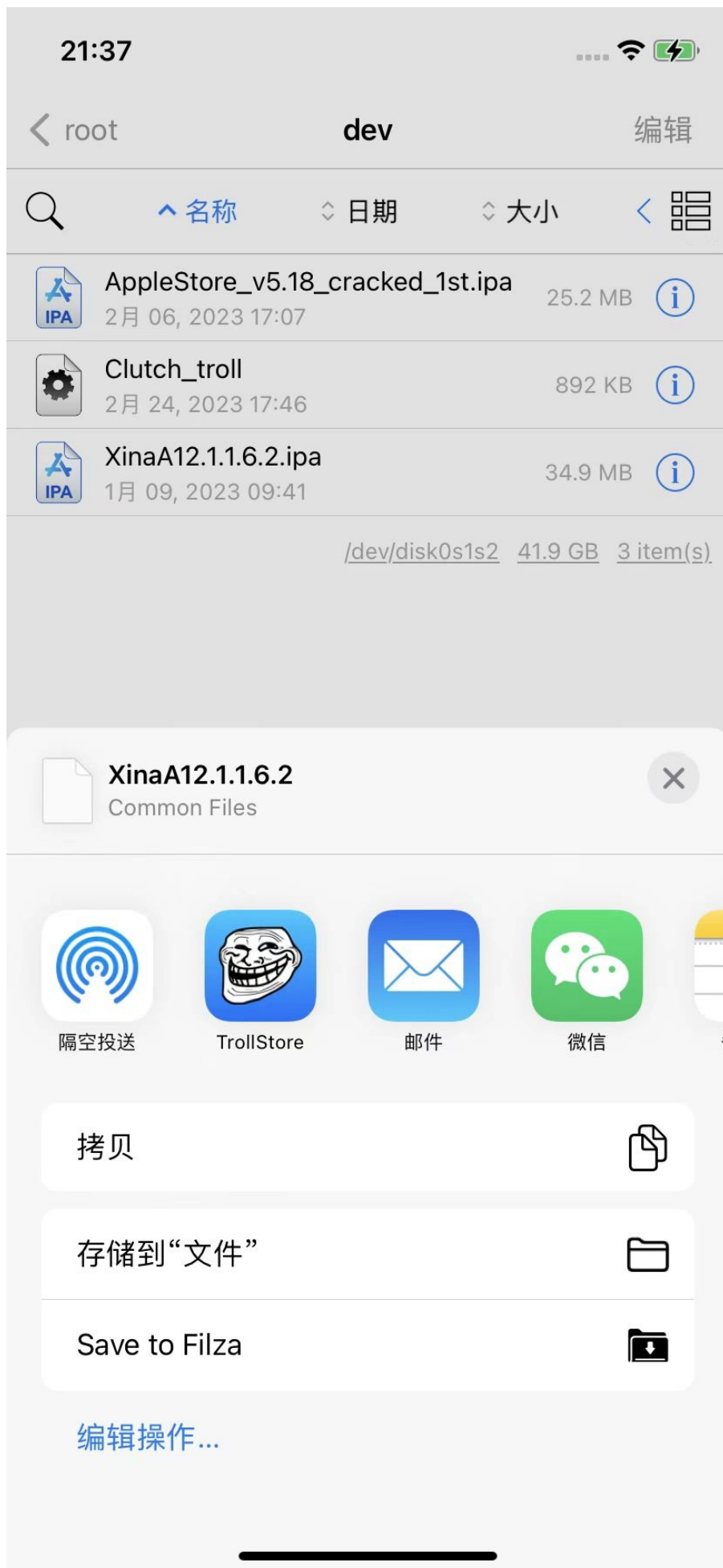


- 长按-》右键菜单





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



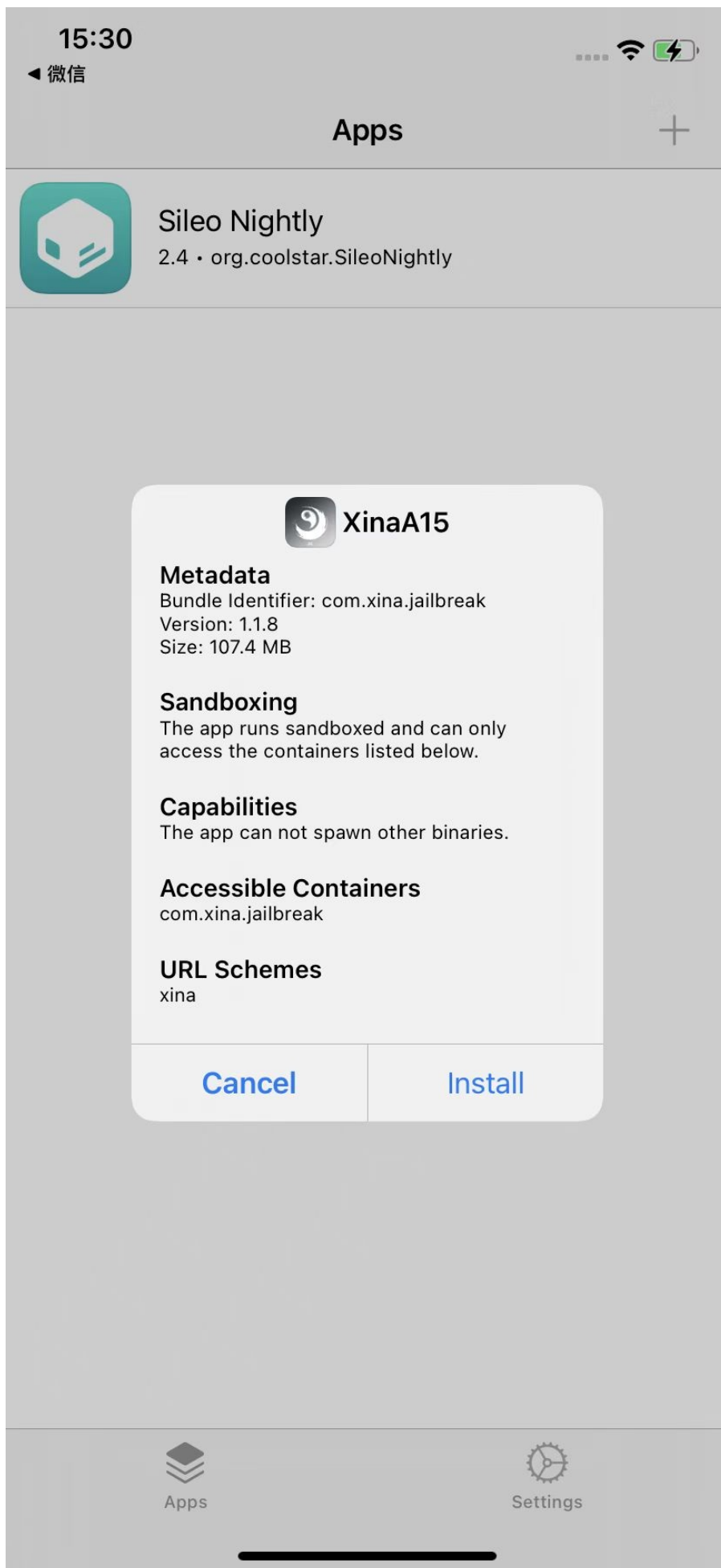
- 方式2: 选择 打开方式 -> 选择 TrollStore

-

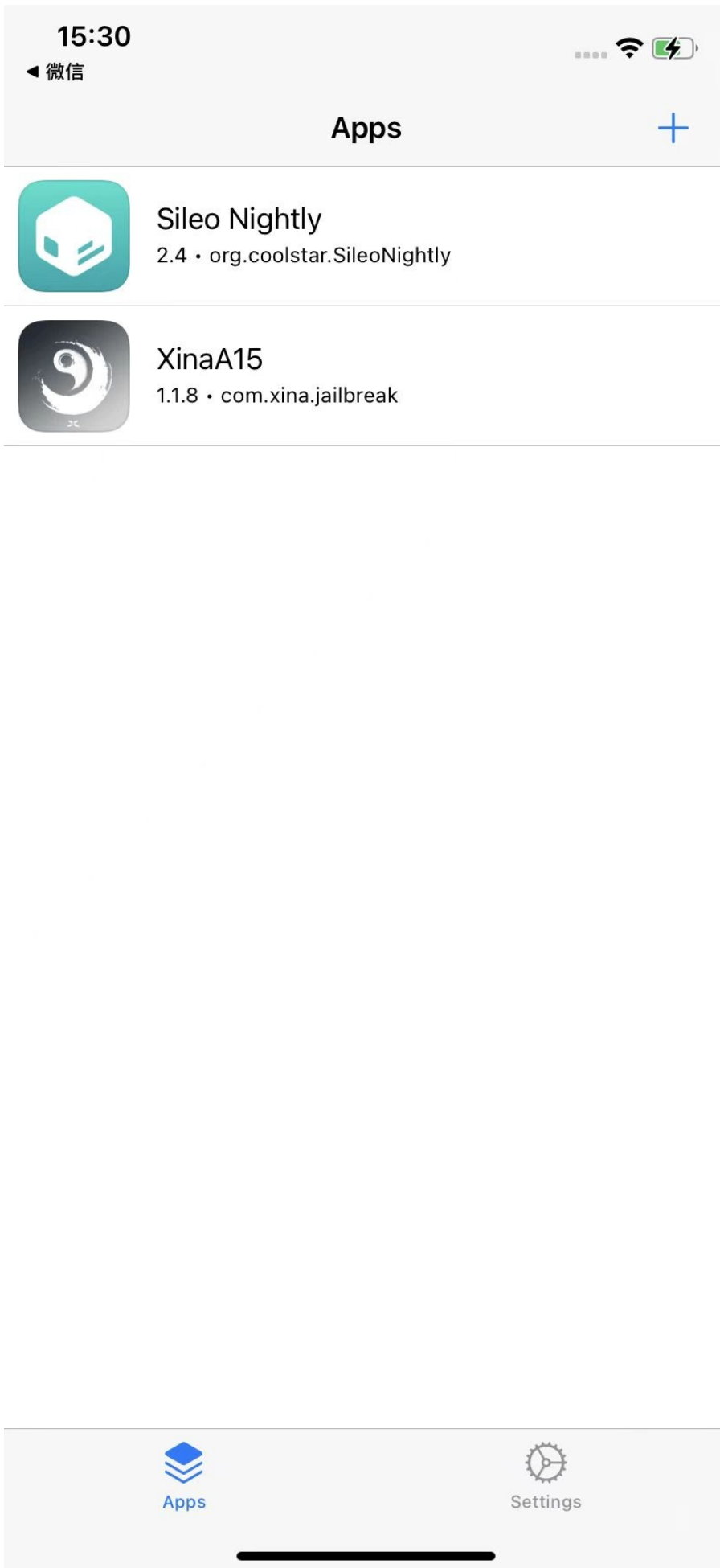


用TrollStore打开并自动安装XinaA15的ipa文件

- 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`
 -

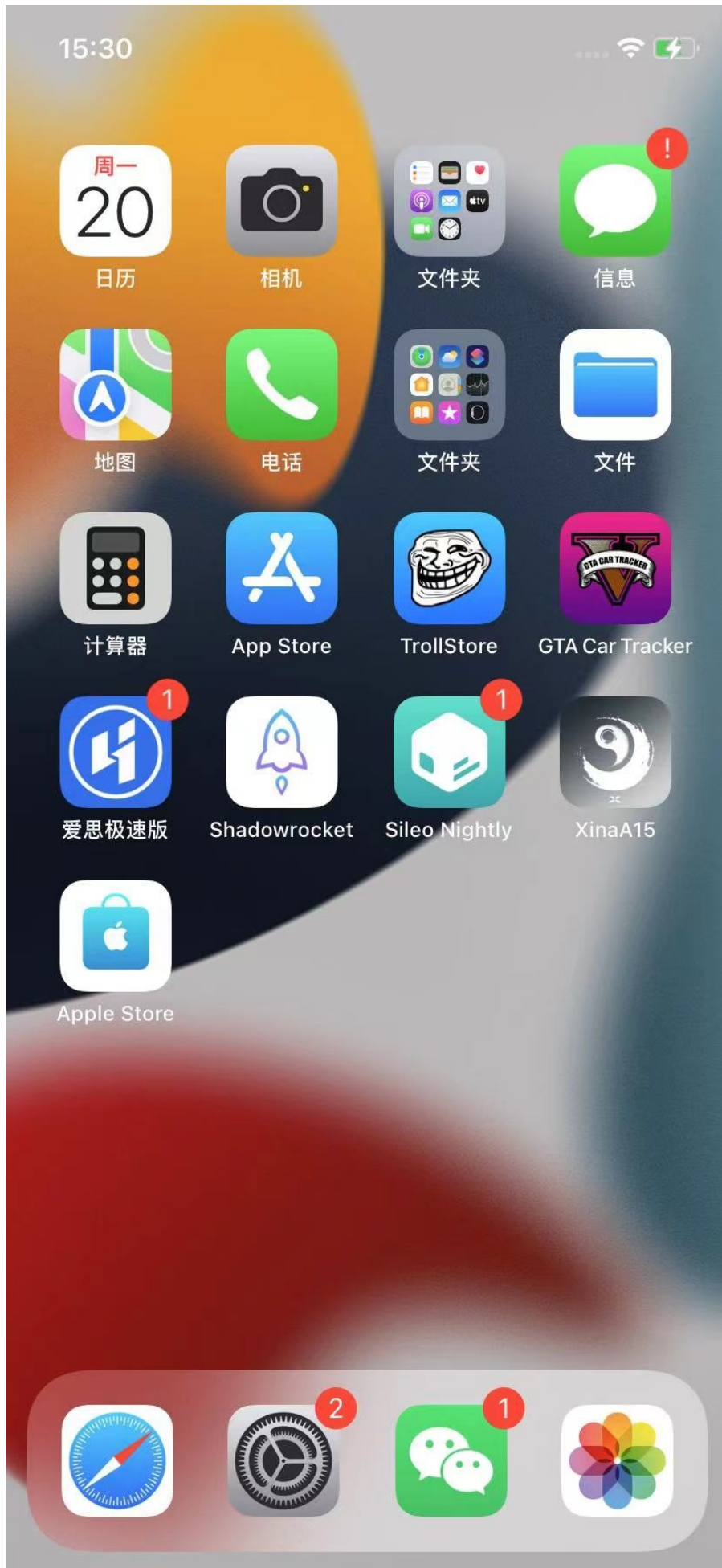


- 显示Installing
- 安装完毕后，即可在 Apps tab页面中看到已安装的 XinaA15
 -



- 安装后的桌面出现XinaA15的图标了

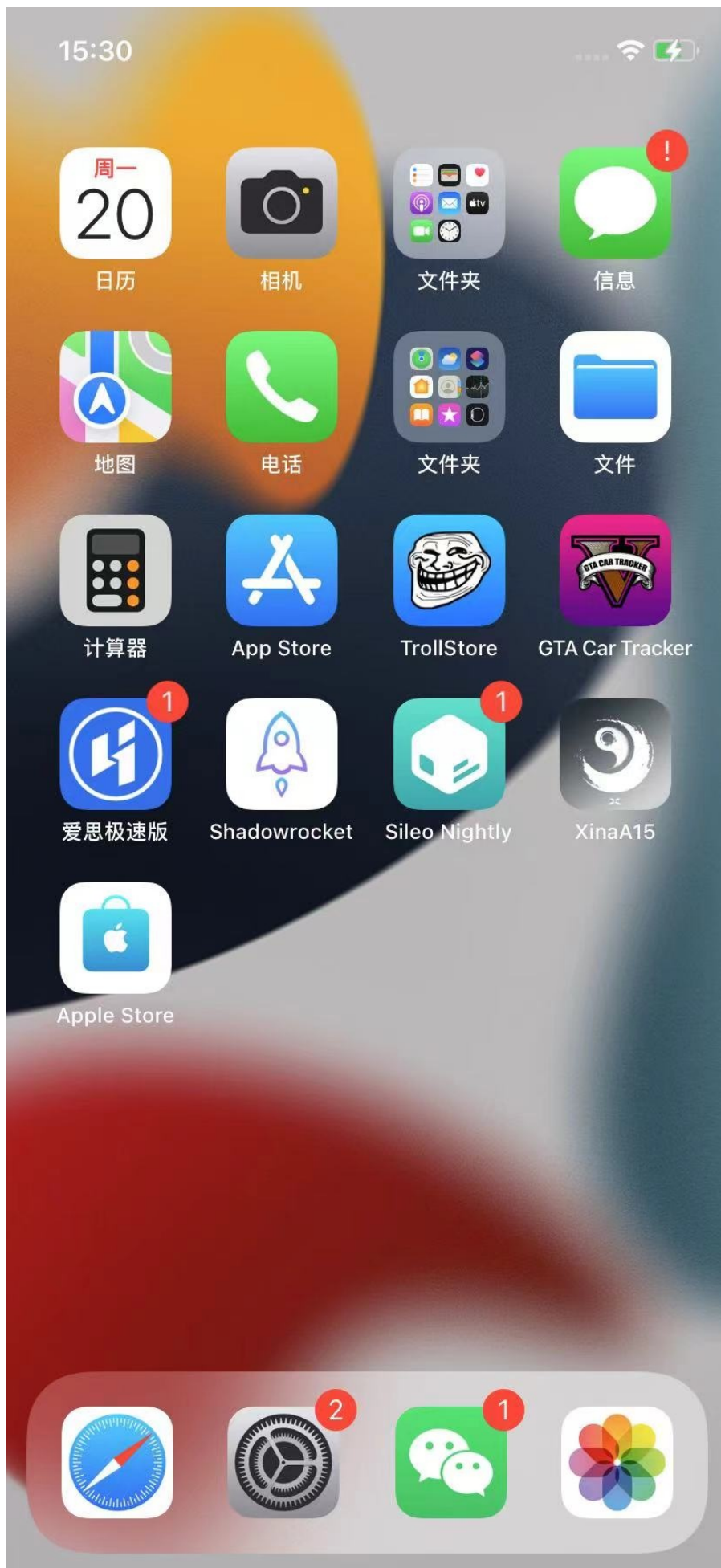




用XinaA15越狱=开启越狱

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：
 -



- 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
 -

设置 注销 软重启 进入临退 退出

```

link指向文件已修复 0 /usr/bin/zfgrep /private/preboot/
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=ffffff009ed8a60:
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



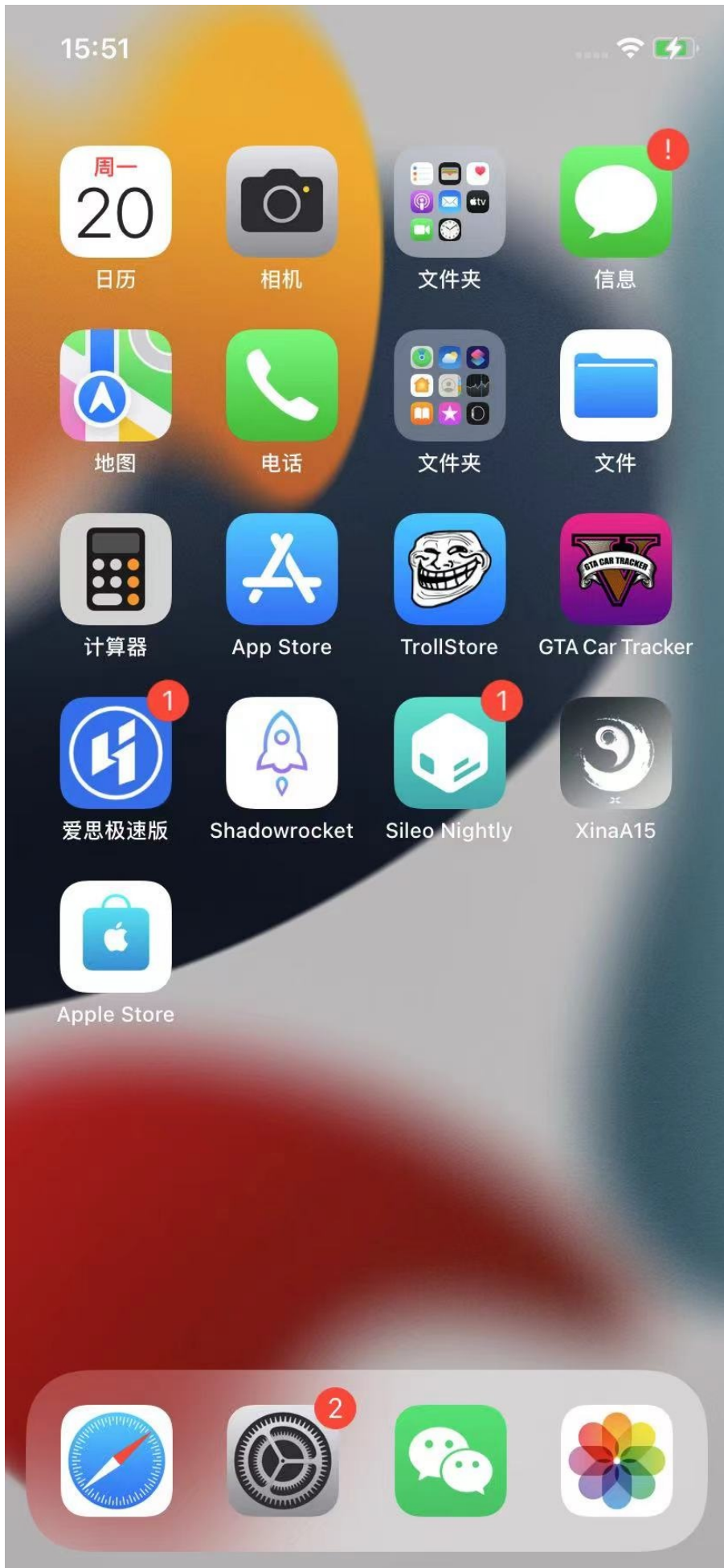
内核内存

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
 -



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

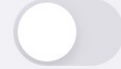
语言 (Lang...)

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

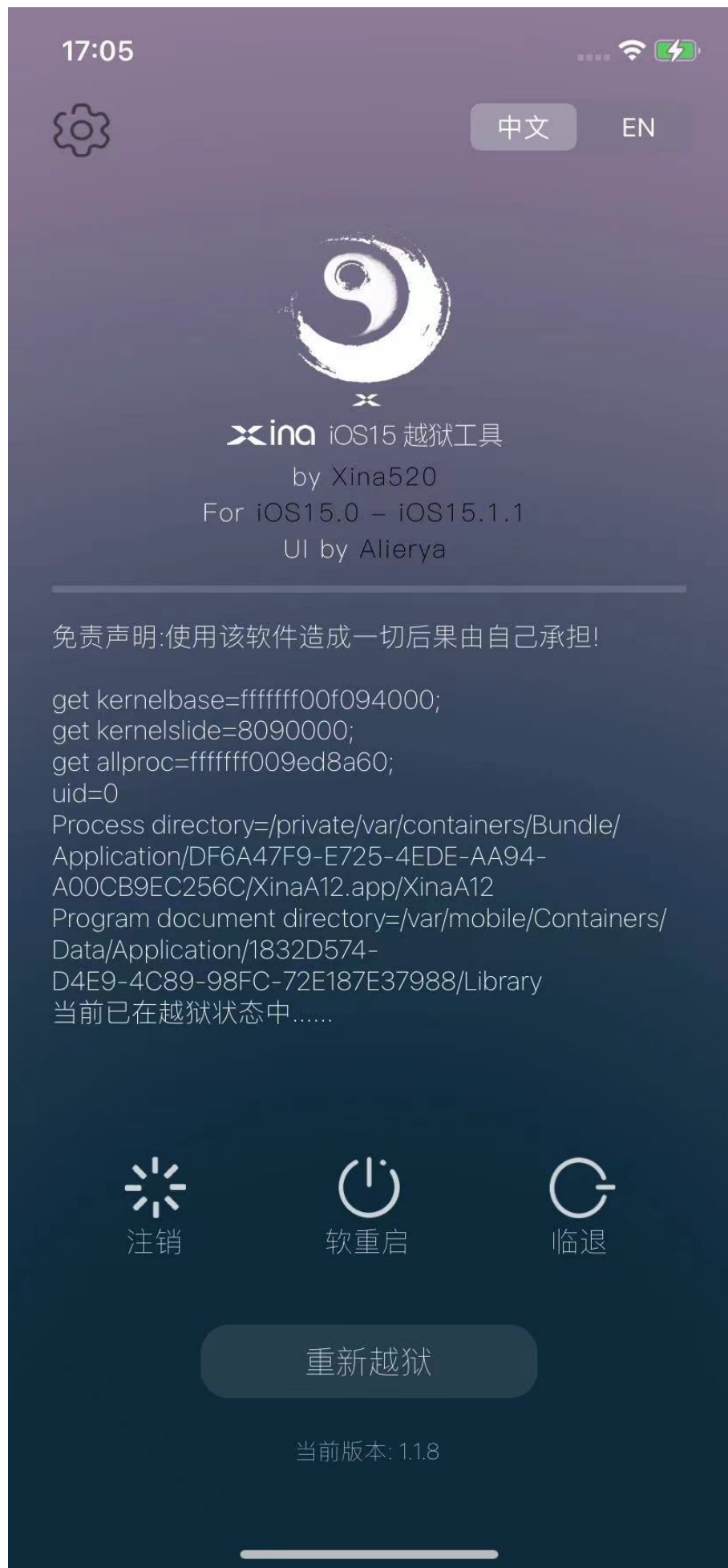
XinaA15本身的使用

(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
 -



- 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
 -



- 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

-

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 已存在!
重启 jailbreakd
jailbreakd 服务已启动...
launchhook 已存在! 重启 launchhook 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
 -



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





- 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

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2023-06-28 22:46:49

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=ffffff009ed8a60:  
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/XPC Services/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



0xFFFFFFFF

内存查看

回退



越狱



文件管理器

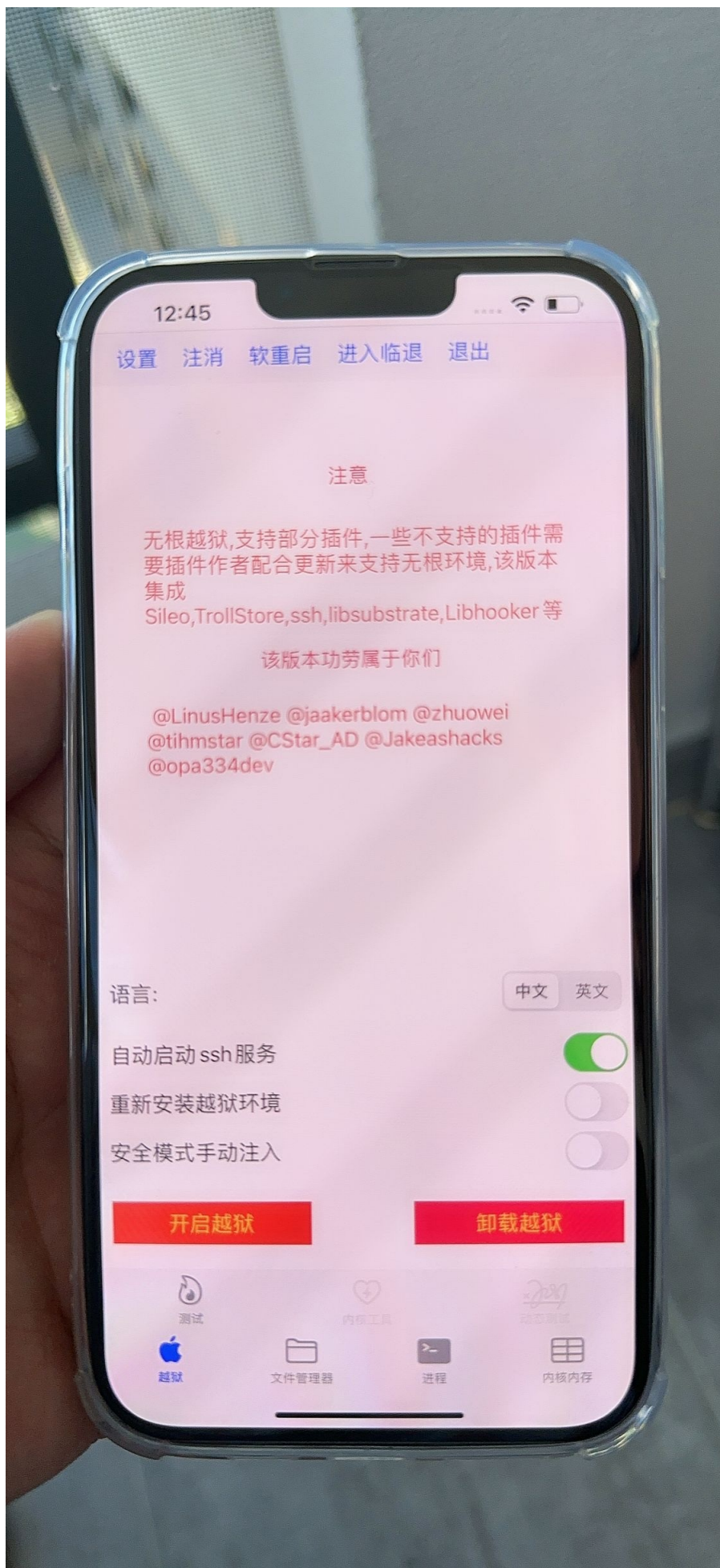


进程



内核内存

- 更早版本的界面
 -



XinaA15 1.1.8之后：新的UI界面

- 1.1.8
 - 主界面
 -

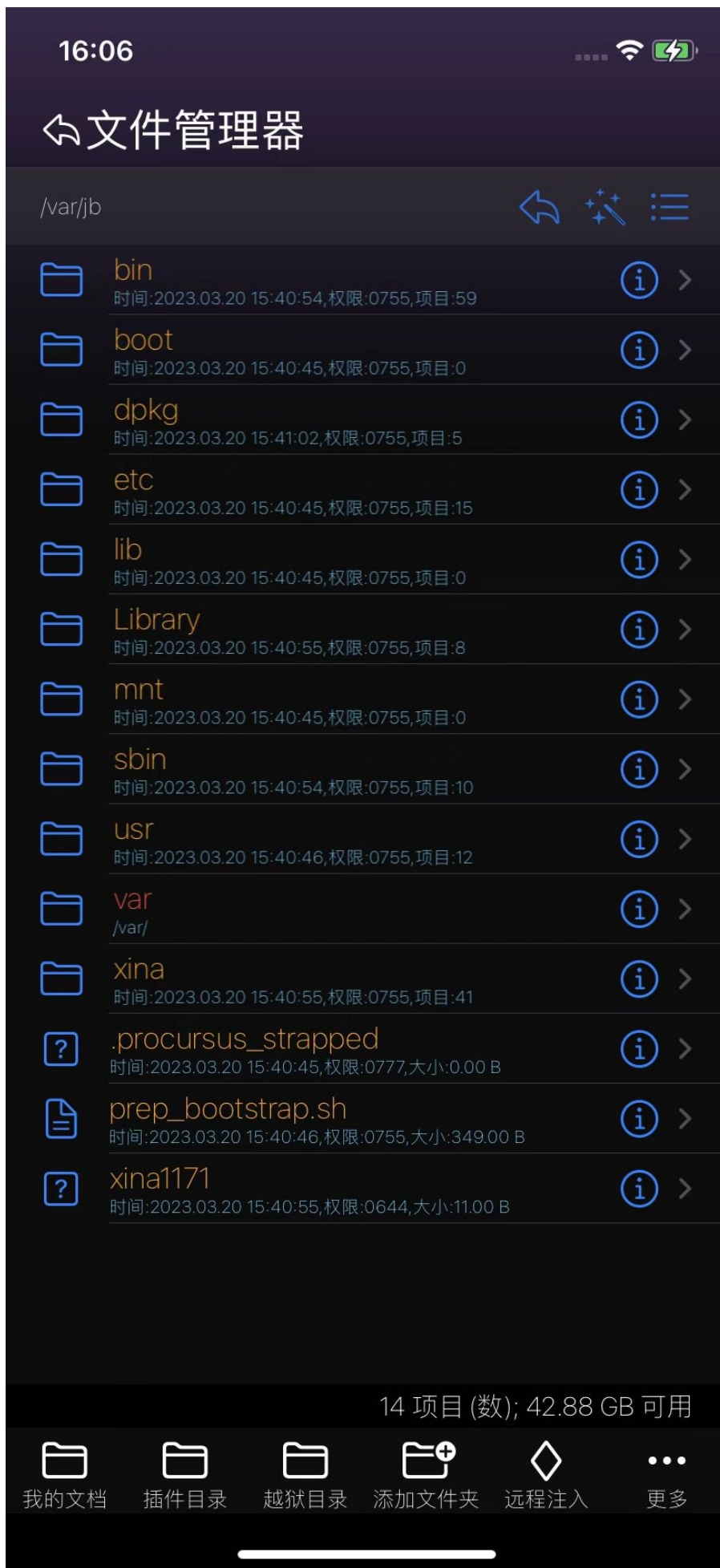


- 设置



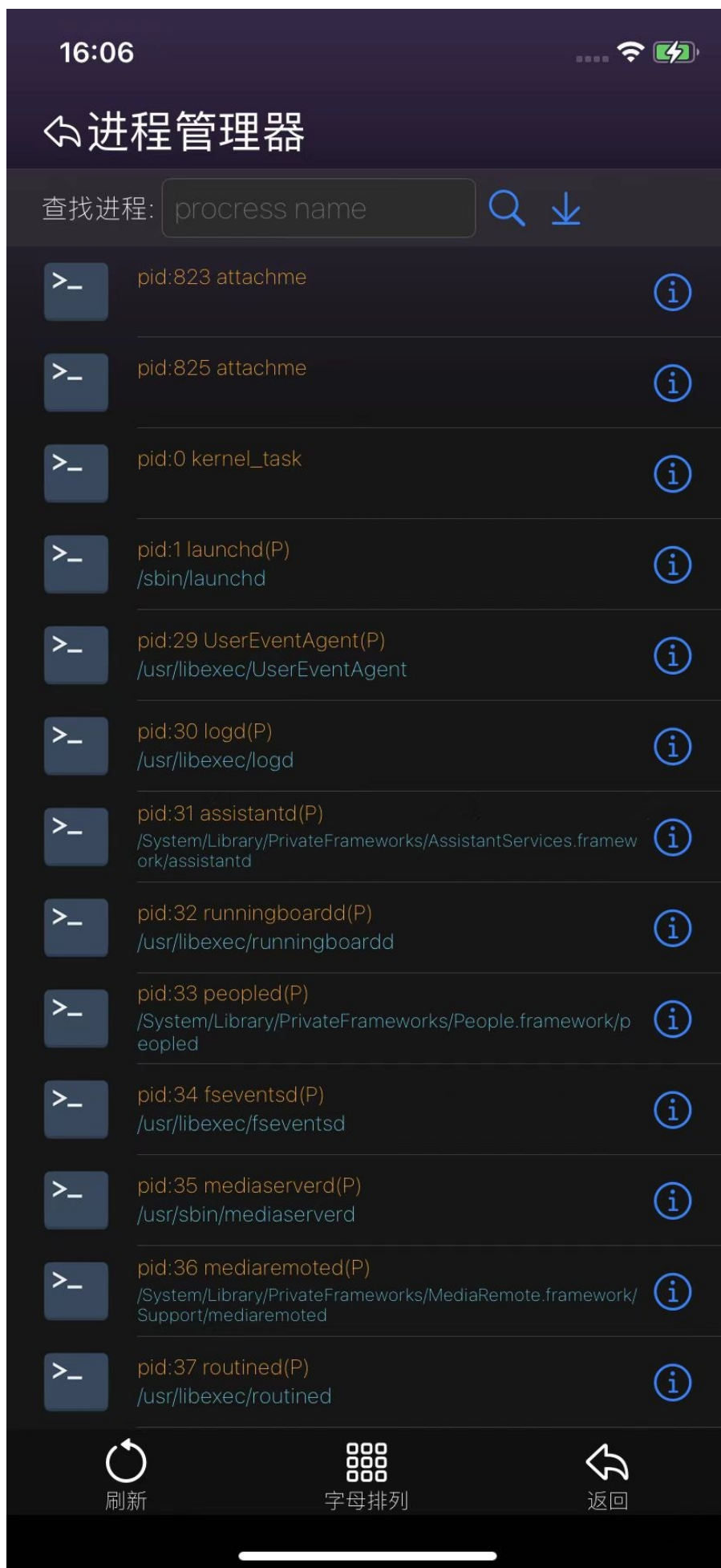


- 其他功能模块
 - 文件管理器
 -



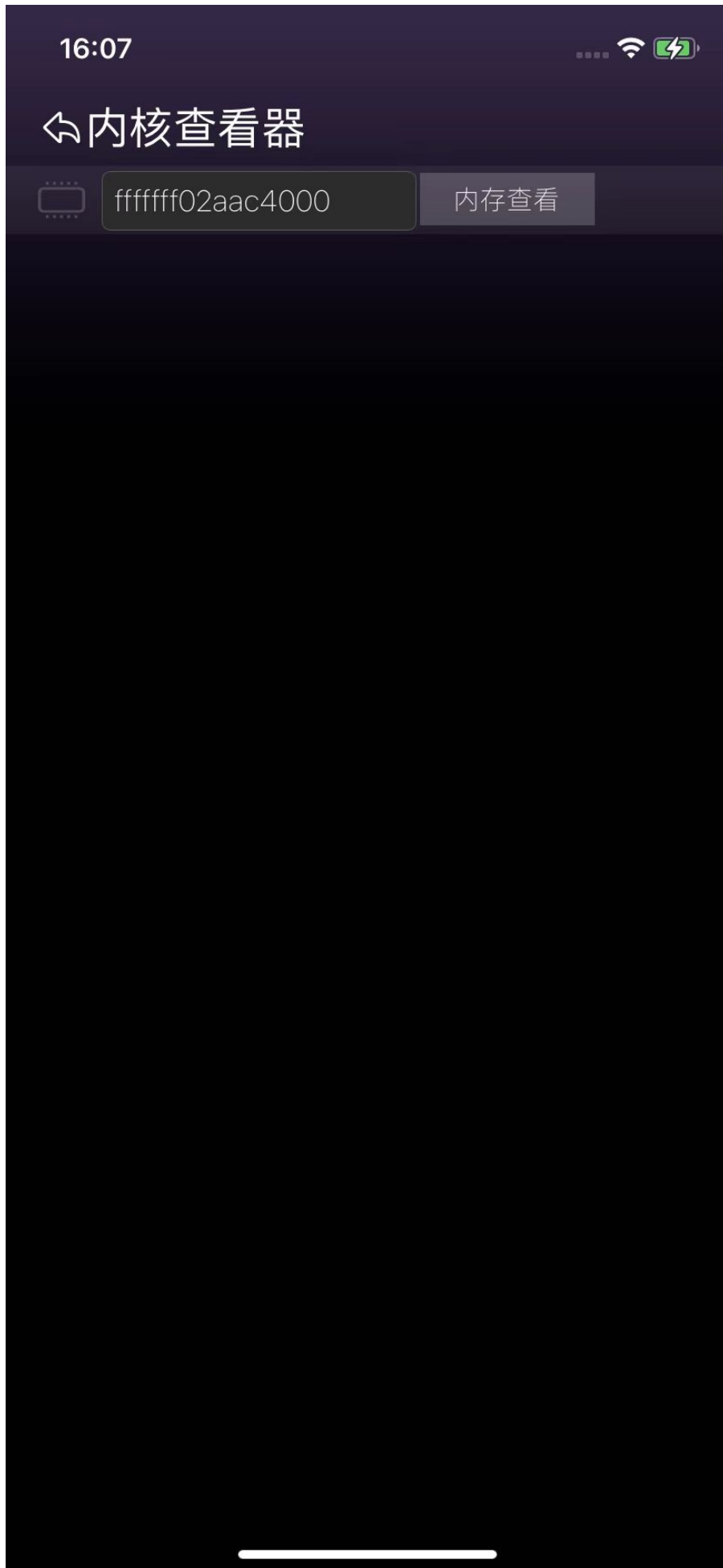
- 进程管理器

-



- 内核查看器

-



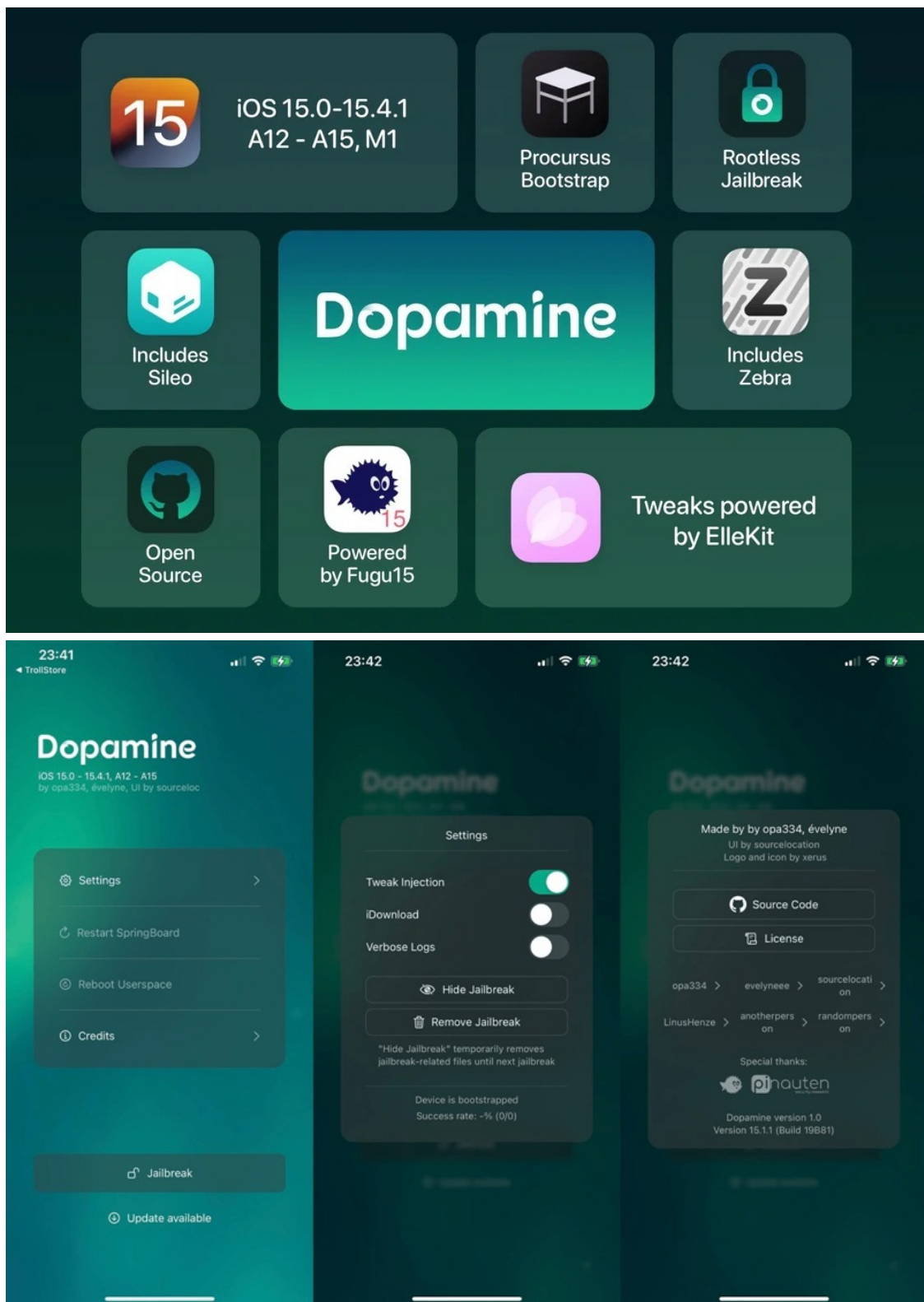
Dopamine

- 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

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2023-06-28 23:03:12

附录

下面列出相关参考资料。

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2023-06-28 21:16:19

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procur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Idid报错: Error downloading Idid 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】iSOpenDev的插件dylib注入iPhone11失败: mach-o file but is an incompatible architecture have arm64 need arm64e
-

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2023-07-10 15:00:10