

# 目录

前言	1.1
iPhone越狱概述	1.2
越狱前	1.3
名词解释	1.3.1
越狱工具	1.3.2
unc0ver	1.3.2.1
checkra1n	1.3.2.2
palera1n	1.3.2.3
XinaA15	1.3.2.4
iPhone机型信息	1.3.3
越狱中	1.4
给iPhone越狱	1.4.1
unc0ver	1.4.1.1
checkra1n	1.4.1.2
palera1n	1.4.1.3
常见问题	1.4.1.3.1
help语法	1.4.1.3.2
loader app	1.4.1.3.3
XinaA15	1.4.1.4
越狱后	1.5
palera1n	1.5.1
恢复越狱	1.5.2
文件管理	1.5.3
包管理器	1.5.4
Cydia	1.5.4.1
插件	1.5.4.1.1
Apple File Conduit "2"	1.5.4.1.1.1
AppSync Unified	1.5.4.1.1.2
OpenSSH	1.5.4.1.1.3
Filza	1.5.4.1.1.4
iCleaner Pro	1.5.4.1.1.5
Mterminal	1.5.4.1.1.6
使用心得	1.5.4.1.2
Sileo	1.5.4.2
爱思助手	1.5.5
安装ipa	1.5.6
附录	1.6
参考资料	1.6.1



# iOS逆向开发：iPhone越狱

- 最新版本: v1.1
- 更新时间: 20230309

## 简介

iOS逆向开发系列教程之iPhone越狱，先是越狱的概述，再介绍越狱前要选择合适越狱工具，比如unc0ver、checkra1n、palera1n、XinaA15等；以及越狱中，如何用unc0ver、checkra1n、palera1n等工具给iPhone手机越狱，以及越狱后的各种事项，包括恢复越狱、文件管理、包管理的Cydia和Sileo、辅助工具爱思助手和安装ipa。

## 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\\_iphone\\_jailbreak: iOS逆向开发：iPhone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逆向开发：iPhone越狱 book.crifan.org](#)
- [iOS逆向开发：iPhone越狱 crifan.github.io](#)

### 离线下载阅读

- [iOS逆向开发：iPhone越狱 PDF](#)
- [iOS逆向开发：iPhone越狱 ePUB](#)
- [iOS逆向开发：iPhone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： 2023-03-09 23:13:10

# iPhone越狱概述

- 给iPhone越狱
  - iOS 13
    - 常用越狱工具
      - checkra1n
      - unc0ver
    - 核心越狱步骤
      - 可以直接手动用工具（unc0ver、checkra1n等）去越狱
      - 也可以借助爱思助手去越狱
      - 爱思助手里面有个一键越狱集成了很多方便的工具，简化了越狱过程
  - iOS 15+
    - iOS 15.0 - iOS 15.1.1
      - XinaA15
    - iOS 15.0 ~ iOS 16.3.1
      - palera1n
- 越狱后
  - 文件管理
    - 爱思助手的文件管理
    - ssh登录
      - scp通过ssh拷贝
    - Filza文件管理器

---

TODO:

- 【整理】iOS的iPhone越狱和改机相关知识
- 【已解决】Activator是什么
- 
- 【无法解决】iPhone X用爱思助手的unc0ver越狱失败：正在安装unc0ver越狱，失败
- 【无法解决】用爱思助手通过unc0ver给iPhone X越狱
- 【已解决】手动用unc0ver去给iPhone X越狱
- 【已解决】Mac中用Cydia Impactor去安装unc0ver到iPhone X
- 【已解决】iPhone X中用unc0ver越狱
- 【已解决】iPhone X用unc0ver越狱后越狱失败爱思助手仍显示未越狱
- 【已解决】Cydia Impactor安装unc0ver的ipa报错：file provision.cpp what Please sign in with an app-specific password
- 【已解决】生成Apple ID的app专用密码
- 【记录】给iPhone X初始化准备越狱开发环境

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：2023-03-02 21:54:22

# 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: 2022-06-21 13:51:47

## 名词解释

越狱前，要了解很多基本概念：

**jb = jailbreak = 越狱**

- 技术角度
  - 越狱=iOS越狱=iOS jailbreak = iPhone越狱
    - 含义：获取iOS设备的Root权限的技术手段
- 类比：iPhone=iOS系统，就像一个监狱
  - 普通iPhone的用户，就像在监狱内，虽然被iOS系统管理约束着，也很安全，但是失去了很多自由
  - 想要更加自由，就要从监狱中逃出来 = 越狱
  - 摆脱iOS系统的约束，拥有更多自由
  - 可以安装更多更好的插件、应用等，做之前非越狱时的不能做的各种事情

## 普通越狱 vs rootless无根越狱

- 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` = `rootful` 越狱
- `rootless` 越狱 = 无根越狱
  - rootfs只读，只有/var可写

## 完美越狱 vs 不完美越狱 vs 半完美越狱 vs 半不完美越狱

- 核心逻辑：2种类型
  - 重启后，是否要（用特殊软件）引导才能开机
    - 是
      - 全越狱=完美越狱
        - 越狱后的iPhone可以正常关机和重启
    - 否
      - 半越狱=不完美越狱
        - iPhone重启后
          - 屏幕就会一直停留在启动画面，也就是“白苹果”状态
          - 或者能正常开机，但已经安装的破解软件都无法正常使用
          - 需要将设备与PC连接后，使用软件进行引导才能使用
  - 重启后，是否还保留越狱状态
    - 是
      - 全越狱
    - 否
      - 半越狱
- 2种核心逻辑组合出4种：越狱类型=Types of Jailbreaks
  - `untethered` = `Untethered jailbreak` = 完美越狱：不需要引导启动
    - 指设备在进行重启后，越狱状态仍被完整保留
  - `semi-untethered` = `Semi-untethered jailbreak` = 半完美越狱
    - 指设备在重启后，将丢失越狱状态；而若想要再恢复越狱环境，只需在设备上进行某些操作即可恢复越狱
  - `tethered` = `Tethered jailbreak` = 不完美越狱：需要引导启动
    - 当处于此状态的iOS设备开机重启后，之前进行的越狱程序就会失效，用户将失去Root权限，需要将设备连接电脑来使用（如特定版本下的红雪（redsn0w））等越狱软件进行引导开机以后，才可再次使用越狱程序。否

则设备将无法正常引导

- semi-tethered = Semi-tethered jailbreak =半不完美越狱
  - 指设备在重启后，将丢失越狱状态，并恢复成未越狱状态。如果想要恢复越狱环境，必须连接计算机并在越狱工具的引导下引导来恢复越狱状态

=> 典型越狱效果：

- 很久之前：越狱后，就一直保持越狱状态（完美越狱）
  - 重启iPhone也能保持越狱状态
- 现在：越狱后，重启iPhone会丢失越狱（半完美越狱）
  - 所以重启iPhone后，往往还要去：恢复越狱
    - 恢复越狱，等于重新执行一遍越狱流程，重新（再次）越狱

## Respring = Reboot SpringBoard =重启桌面=注销

iOS系统内有个默认的，自带的应用：SpringBoard 也就是：你所看到，iPhone的桌面 每次安装越狱插件，为了使得越狱插件生效，则需要，重启桌面，也就是Respring

常见的重启桌面的方式有：

- Filza安装ipa后-》右上角的动作-》注销（就是 Respring）
- 命令行操作：
  - 进入命令行方式
    - Mac中通过ssh进入iPhone的命令行
    - iPhone中通过终端类插件进入命令行
  - 具体命令
    - killall SpringBoard

## uicache =清除界面缓存

有时候，需要清理图标等UI的缓存，桌面上才能看到最新的变化

比如：

- （通过某些工具）安装了app后，uicache后，桌面上才能看到已安装的app的图标
- 删除了app后，由于某些原因，需要uicache后，桌面上的app图标才消失 清理UI的cache缓存，有个专门的工具叫做：
- uicache

TODO:

- 【整理】UICache是什么和作用

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：2023-03-02 22:13:49

# 越狱工具

## 确认iPhone信息和版本

- 记录】iPhone6的手机基本信息
- 【记录】Mac中连接iOS 12.4.5的iPhone6

## 越狱工具的选择

- 越狱工具的选择
  - 【整理】iOS iPhone破解和越狱相关的基础知识
  - 【整理】不同版本iOS系统的越狱工具的选择
  - 【整理】ios 13 越狱工具的选择
  - 【整理】iOS越狱工具对比：checkra1n、unc0ver、Electra、Pangu等
  - 【已解决】iOS的半越狱和全越狱
  - 【整理】iOS越狱工具：Pangu盘古
  - 【整理】越狱工具软件：奥德赛 Odyssey

结论：

- iOS <15.0
  - 说明：越狱工具大多都很成熟和稳定，用的人也很多
  - 针对的iOS系统版本
    - 多数都是：iOS 12 ~ iOS 14
    - iOS <12，基本上很少用了
  - 常用
    - checkra1n
    - unc0ver
- iOS >15.0
  - 说明：截至20230302，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
  - 一般用
    - palera1n
    - 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：2023-03-02 22:16:30

# unc0ver

- unc0ver
  - 主页
    - <https://unc0ver.dev/>

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: 2023-03-02 22:19:39

# checkra1n

- checkra1n
  - 主页
    - <https://checkra.in>

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: 2023-03-02 22:19:08

# 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
      - iPhone 11
      - iPhone 11 Pro
      - iPhone 11 Pro Max
      - iPhone 12
      - iPhone 12 Pro
      - iPhone 12 Pro Max
      - iPhone 13
      - iPhone 13 Pro
      - iPhone 13 Pro Max
      - 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)
      - iPad (8th generation)
      - iPad (9th generation)
      - iPad mini (5th generation)
      - iPad Air (3rd generation)
      - iPad Pro (11")
      - iPad Pro (12.9") (3rd generation)
      - iPad (10th generation)
      - iPad (11th generation)
      - iPad (12th 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越狱的相关说明

- 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 = rootFull`
        - `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>

## 参考资料

- [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 \(github.com\)](#)

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: 2023-03-09 22:59:16

# XinaA15

- XinaA15
  - 主页
    - <https://xina.ss03.cn/>
  - 下载
    - 截至20230302, 最新版是: 1.1.7.1
      - 三种安装方式
        - TrollStore
          - apple-magnifier://install?url=https://apt.xina.vip/XinaA12.1.1.7.1.ipa
        - 在线
          - <https://www.lanzouy.com/io7et0ot5omf>
        - IPA
          - <https://apt.xina.vip/XinaA12.1.1.7.1.ipa>

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: 2023-03-02 22:22:30

# 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。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

# 越狱中

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: 2022-06-21 13:49:28

# 给iPhone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: 2023-03-02 22:22:52

## unc0ver

TODO:

- 【已解决】iPhone中用unc0ver去给iOS越狱
  - 【记录】用unc0ver重新恢复越狱后的iPhone中的效果
  - 【已解决】给用unc0ver越狱的iPhone7恢复越狱状态
  - 【记录】unc0ver越狱iPhone7的完整log日志
  - 【记录】用unc0ver还原卸载越狱环境
-

## checkra1n

TODO:

- 【已解决】Mac中给iOS 12.4.5的iPhone6中安装checkra1n
  - 【研究】用unc0ver越狱iOS 13的iPhone
  - 【已解决】Mac中用checkra1n越狱iOS 12.4.5的iPhone6
-

## palera1n

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 -c
    - 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>

The screenshot shows the GitHub releases page for the palera1n-macos-universal asset. The page header includes the URL <https://github.com/palera1n/palera1n-c/releases>. Below the header, there is a list of available assets:

Asset	Size	Last Updated
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), checkra1n team (Siguza, axi0mx, littlelailo
# et al.), Procurus 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.), Procurus 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 abdc0dd961c3cb96f5c4afe109de4eb48b88433a to enter recovery mode immediately
- [02/27/23 14:13:20] Info : Press Enter when ready for DFU mode
```

```

65% ~ 10 GB 8.2 kB↓ 39 kB↑ 2/27, 14:13
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.), Procurus 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.), Procurus Team (Hayden Seay, Cameron Katri, Keto et.al)

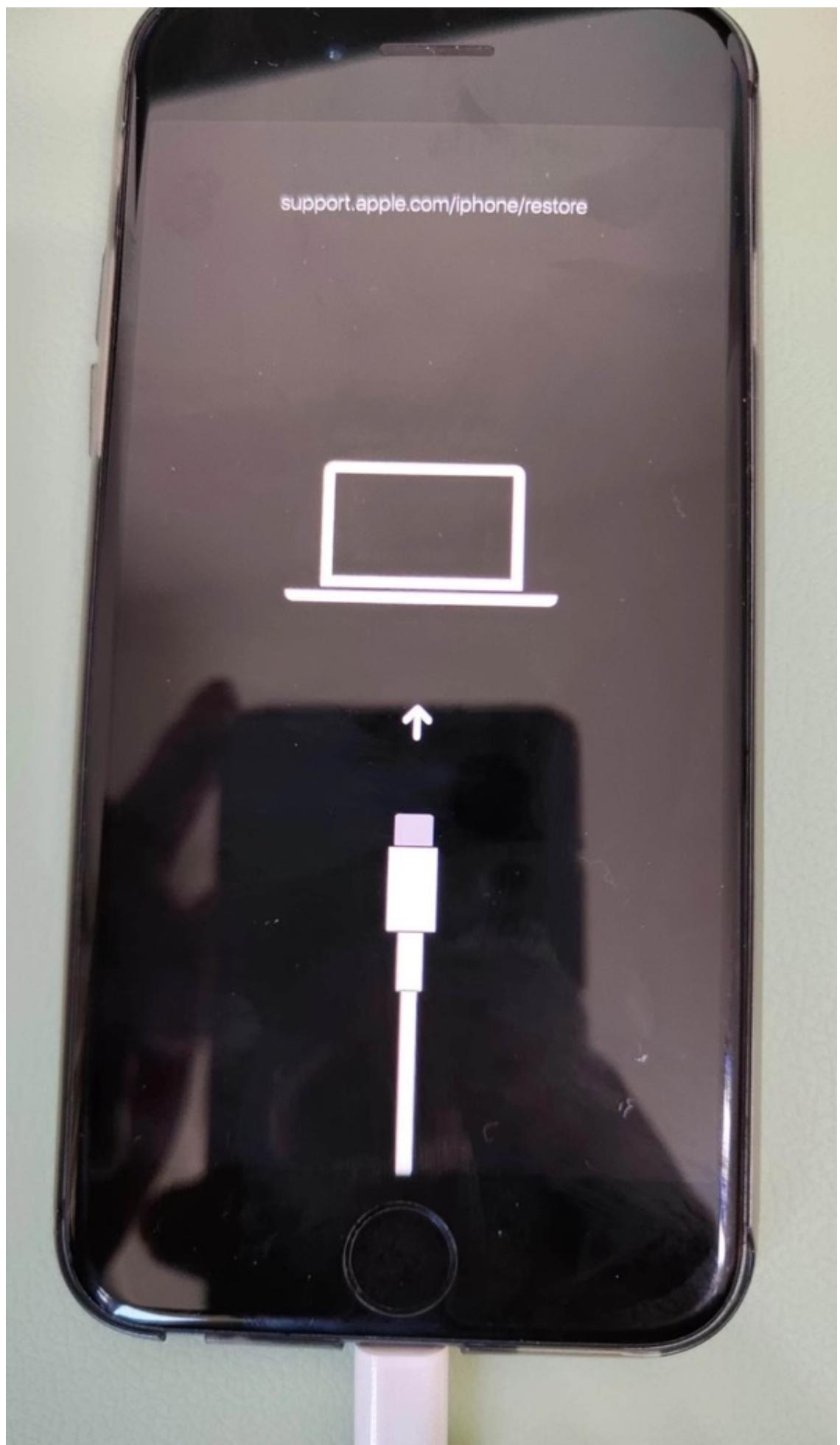
- [ <Info>: Waiting for devices
- [ <Info>: Telling device with uid abdc0dd961c3cb96f5c4afe109de4eb48b88433a to enter recovery mode immediately
- [ <Info>: Press Enter when ready for DFU mode

```

此时：iPhone手机中出现：

- 数据线插入电脑
  - 顶部文字：[support.apple.com/iphone/restore](http://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), checkra1n team (Siguza, axi0mx, littlelailo
# et al.), Procurus 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

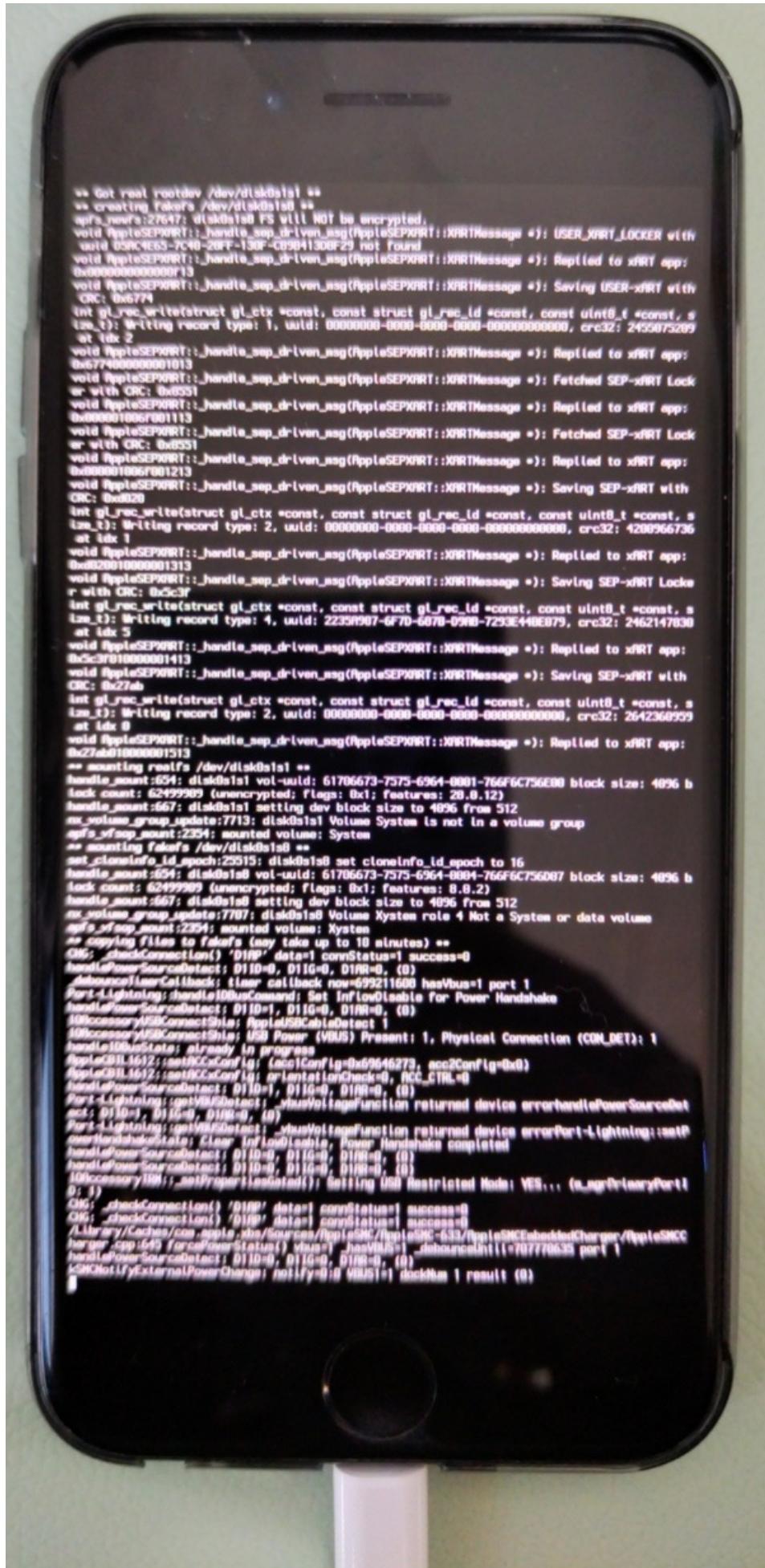
即可继续，进入DFU模式，继续自动越狱过程

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 checkra1n
#
# Checkra1n 0.1337.1
#
# Proudly written in nano
# (c) 2019-2023 Kim Jong Cracks
#
===== Made by =====
# argp, axi0mx, dany1931, jaywalker, kirb, littlelailo, niteTV
# never_released, nullpixel, pimskeks, qwerttyoruiop, sbingner, siguza
===== Thanks to =====
# haifisch, jndok, jONSEALS, xerub, lilstevie, psychotea, sferrini
# Cellebrite (ih8sn0w, cjeri, 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

```

** mounting fakefs /dev/disk0s1d **
set_cloneinfo_id_epoch:25319: disk0s1d set cloneinfo_id_epoch to 16
handle_mount:634: disk0s1d vol->volid: 61786673-7373-6964-0004-766F6C736007 block size: 4096 b
lock count: 6299989 (unencrypted); flags: 0x1 features: 0x02
handle_mount:667: disk0s1d setting dev block size to 4096 from 512
rx_volume_group_update:7707: disk0s1d Volume Xystem Role 4 Not a System or data volume
ufs_vfsop_mount:2354: mounted volume: Xystem
** copying files to fakefs (may take up to 10 minutes) **
OHG: _checkConnection() 'DIRP' data=1 connStatus=1 success=0
handlePowerSourceDetect: D1ID=0, D1IG=0, D1RR=0, (0)
debounceTimerCallback: timer callback now=699211600 hasVbus=1 port 1
Port-Lightning: handleIDBusCommand: Set InflowDisable for Power Handshake
handlePowerSourceDetect: D1ID=1, D1IG=0, D1RR=0, (0)
10AccessoryUSBConnectShts: AppleUSB Cable Detect
10AccessoryUSBConnectShts: USB Power (VBUS) Present: 1, Physical Connection (CON_DETACH) 0
handleIDbmState: already in progress
AppleCBL1612::setICCConfig: (acc1Config=0x69646273, acc2Config=g=0x0)
AppleCBL1612::setICCConfig: orientationCheck=0, RCC_CTRL=0
handlePowerSourceDetect: D1ID=1, D1IG=0, D1RR=0, (0)
Port-Lightning::getVBUSDetect: _vbusVoltageFunction returned device error handlePowerSourceDetect: D1ID=1, D1IG=0, D1RR=0, (0)
Port-Lightning::getVBUSDetect: _vbusVoltageFunction returned device error Port-Lightning::setPortOverHandshakeState: Clear InflowDisable, Power Handshake completed
handlePowerSourceDetect: D1ID=0, D1IG=0, D1RR=0, (0)
handlePowerSourceDetect: D1ID=0, D1IG=0, D1RR=0, (0)
10AccessoryIRRM::setPropertiesGated(): Setting USB Restricted Mode: YES... (m_mprPrimaryPort 0: 1)
OHG: _checkConnection() 'DIRP' data=1 connStatus=1 success=0
OHG: _checkConnection() 'DIRP' data=1 connStatus=1 success=0
/Libraries/Caches/com.apple.xbs/Sources/AppleSNC/AppleSNC-633/AppleSNCEmbeddedCharger/AppleSNC
Charger.cpp:645 ForcePowerStatus() vbus=1 _hasVbus=1 _debounceUntil=707778635 port 1
handlePowerSourceDetect: D1ID=0, D1IG=0, D1RR=0, (0)
ASNDNotifyExternalPowerChange: notify=0:0 VBUS=1 dockNum 1 result (0)
ENo2:client11 New BlogBuf tow 7
this2:client11 New BlogBuf tow 6
virtual bool CoreAnalyticsPipe::checkForWork():174:starting CoreAnalyticsMessenger
void CoreAnalyticsHub::handleNagTimerExpiry(IOTimerEventSource *):627:messageClients of 3 av
able events
ESNCcharger.cpp:1954) Not charging:0
txFlush:1171: disk0s1 xid 5555239 tx stats: # 28 finish 2 enter 7955 wait 17 5299us close 0
txFlush:1171: disk0s1 xid 5555239 tx stats: # 48 finish 3 enter 79134 wait 35 5759us close 0
txFlush:1171: disk0s1 xid 5555239 tx stats: # 48 finish 3 enter 79134 wait 35 5759us close 0
** unmounting filesystems **
ufs_vfsop_umount:2565: disk0s1s1 disk0s1: unmounting volume
ufs_vfsop_umount:2590: disk0s1s1 waiting for purgatory cleaner to finish
ufs_vfsop_umount:2592: disk0s1 rx_nme_vols_mounted is 0
ufs: total mem allocated: 45271927 (43 mb);
ufs_vfsop_umount:2945: all done, going home. (numMountedRPFSThree 0)
ufs_vfsop_umount:2565: disk0s1s0 disk0s1: unmounting volume
ufs_vfsop_umount:2590: disk0s1s0 waiting for purgatory cleaner to finish
rx_volume_group_update:7707: disk0s1s0 Volume Xystem role 4 Not a System or data volume
ufs_vfsop_umount:2932: disk0s1 rx_nme_vols_mounted is 7
ufs: total mem allocated: 44296370 (42 mb);
ufs_vfsop_umount:2945: all done, going home. (numUnmountedRPFSThree 7)
ufs_stop_bg_work:1065: disk0s1s0 Volume Hardware is unmounting, stop any bg work
umount(/private/var/hardware): Resource busy -- try 'diskutil umount'
ufs_stop_bg_work:1065: disk0s1s7 Volume Update role c0 Not a System or data volume
ufs_vfsop_umount:2565: disk0s1s7 disk0s1: unmounting volume
ufs_vfsop_umount:2590: disk0s1s7 waiting for purgatory cleaner to finish
ufs_vfsop_umount:2932: disk0s1 rx_nme_vols_mounted is 6
ufs: total mem allocated: 44296970 (42 mb);
ufs_vfsop_umount:2945: all done, going home. (numUnmountedRPFSThree 6)
ufs_stop_bg_work:1065: disk0s1s7 Volume Basband Data is unmounting, stop any bg work
ufs_vfsop_umount:2565: disk0s1s7 disk0s1: unmounting volume
ufs_vfsop_umount:2590: disk0s1s7 waiting for purgatory cleaner to finish
ufs_vfsop_umount:2932: disk0s1 rx_nme_vols_mounted is 5
ufs: total mem allocated: 44297582 (42 mb);
ufs_vfsop_umount:2945: all done, going home. (numUnmountedRPFSThree 5)
ufs_stop_bg_work:1065: disk0s1s7 Volume Data is unmounting, stop any bg work
umount(/private/var): Resource busy -- try 'diskutil umount'
ufs_stop_bg_work:1065: disk0s1s7 Volume xBT is unmounting, stop any bg work
umount(/private/xBT): Resource busy -- try 'diskutil umount'
** Done setting up fakefs **
exitpid for '/cores/binpack/bin/sh /cores/crmate_fakesh.sh /dev/disk0s1d 0 1 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：

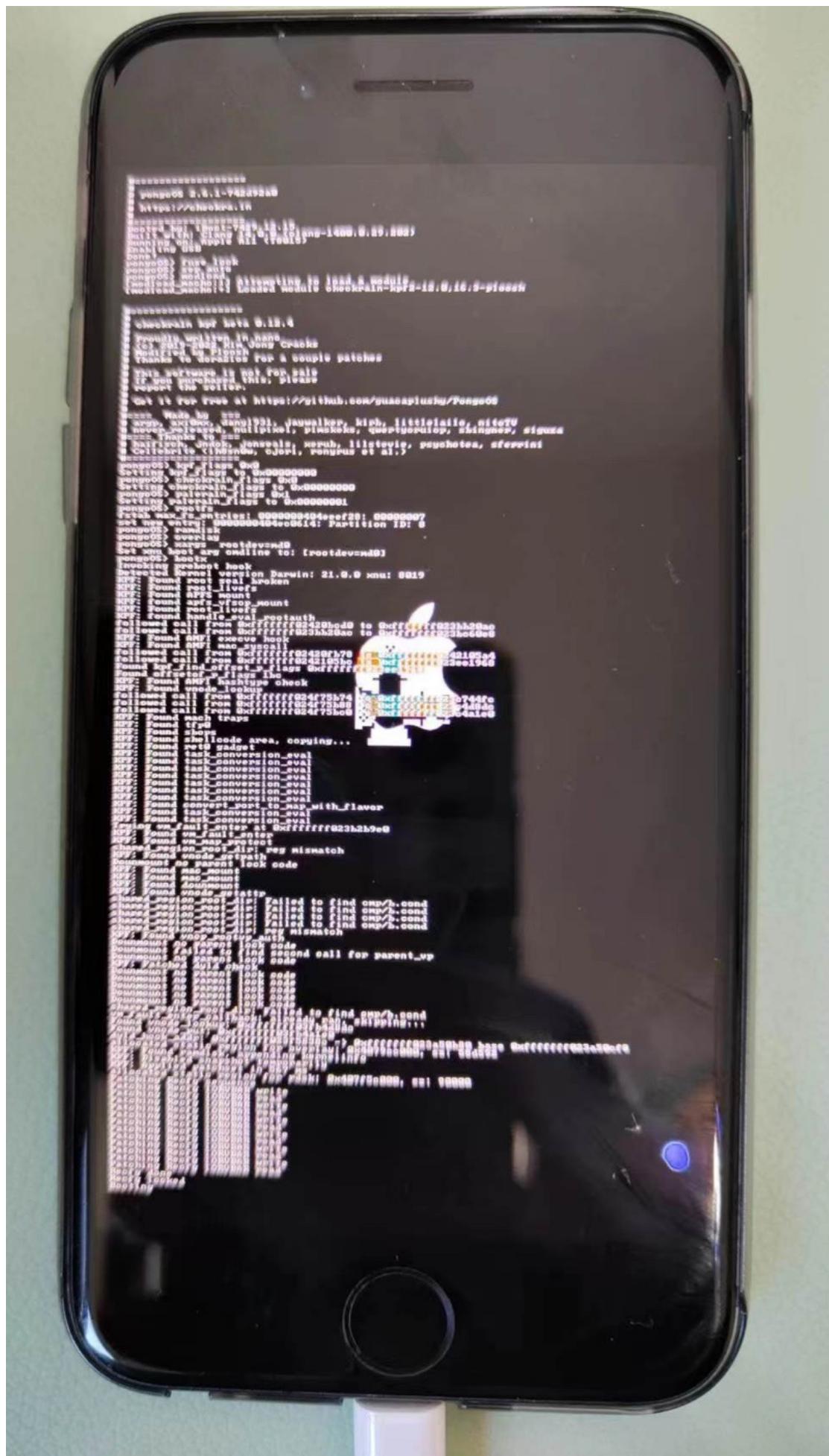
```
crifan@licrifandMacBook-Pro ~ /dev / dev_tool / reverse_security / iOS / palera1n palera1n -f
# == palera1n-c ==
#
# Made by: Nick Chan, Ploosh, Mineek, Nebula, llsc12
#
# Thanks to: dora2ios, pythonplayer, tihmstar, nikias
# (libimobiledevice), checkrain team (Siguza, axi0mx, littlelailo
# et al.), 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 abdc0dd961c3cb96f5c4afe109de4eb48b88433a 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, dany1931, jaywalker, kirb, littlelailo, nitoTV
# never_released, nullpixel, pimskeks, qwertyoruop, sbingner, siguza
===== Thanks to =====
# haifisch, jndok, jonseals, xerub, lilstevie, psychotea, sferrini
# Cellebrate (ih8sn0w, cgori, 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

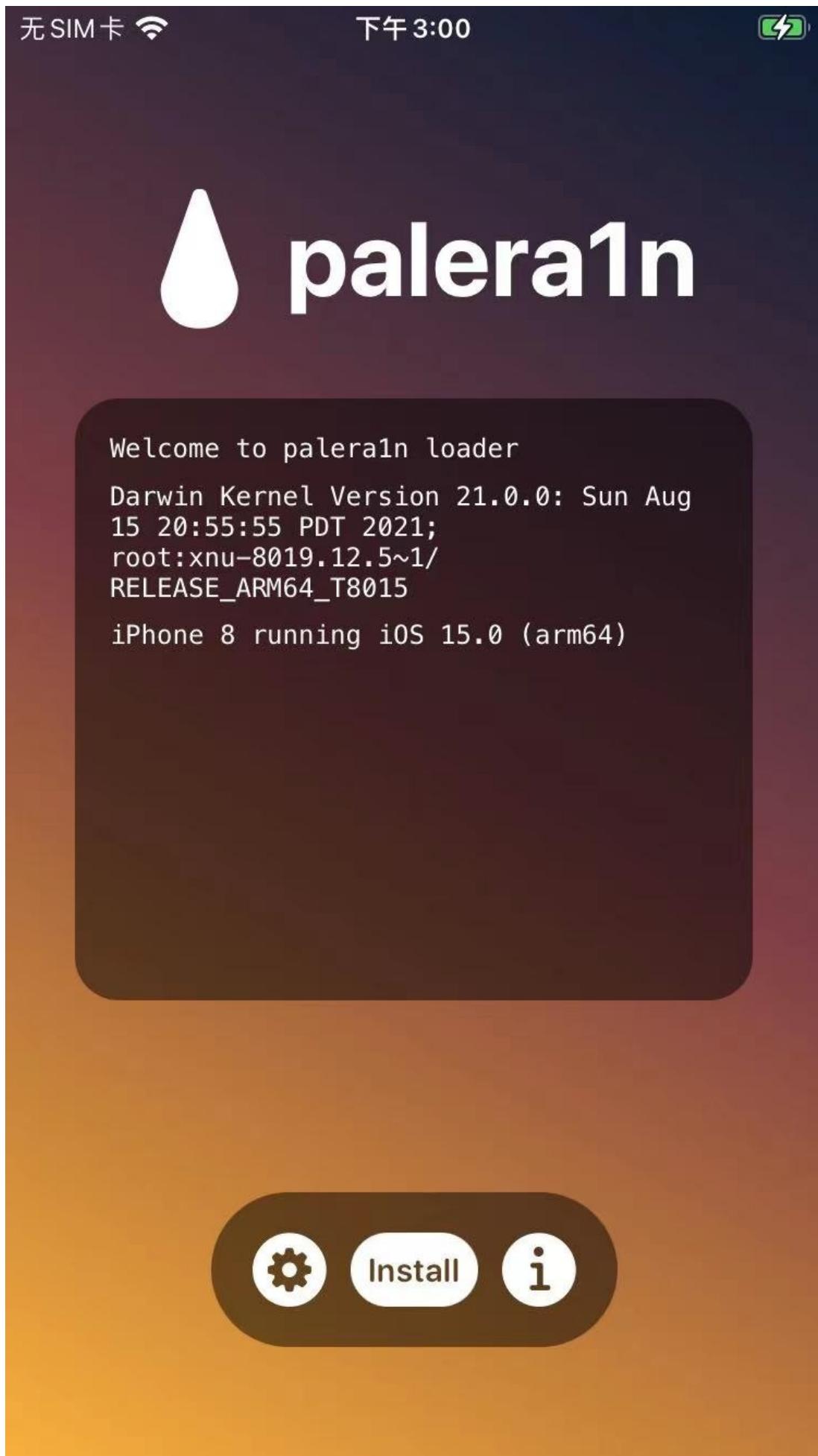
进入iPhone桌面后，可以看到：

palera1n的app = palera1n loader = paleran的图标



#### 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：

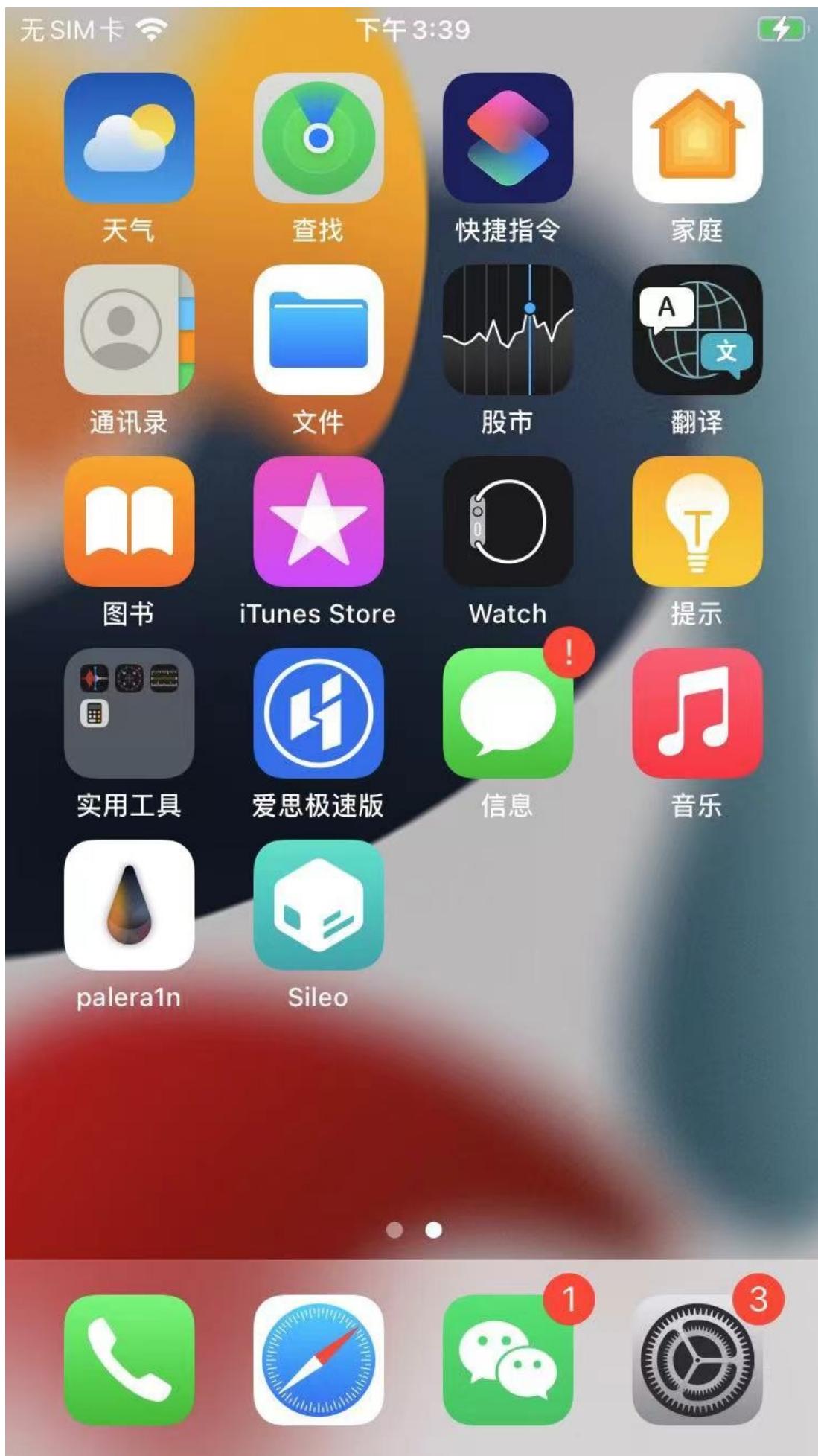
- bootstrap.tar
- sileo.deb
- straprepo.deb

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: 2023-03-09 22:42:59

## 常见问题

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

## palera1n的help语法

```
crifan@licrifandeMacBook-Pro ~/dev/dev_tool/reverse_security/iOS/palera1n palera1n --help
Usage: palera1n [-cCdDEfh1LnOpRsvV] [-e boot arguments] [-k Pongo image] [-o overlay file] [-r ramdisk file] [-K KPF file] [-i checkrain 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-checkrain file  Override checkrain
-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 checkrain 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)
```

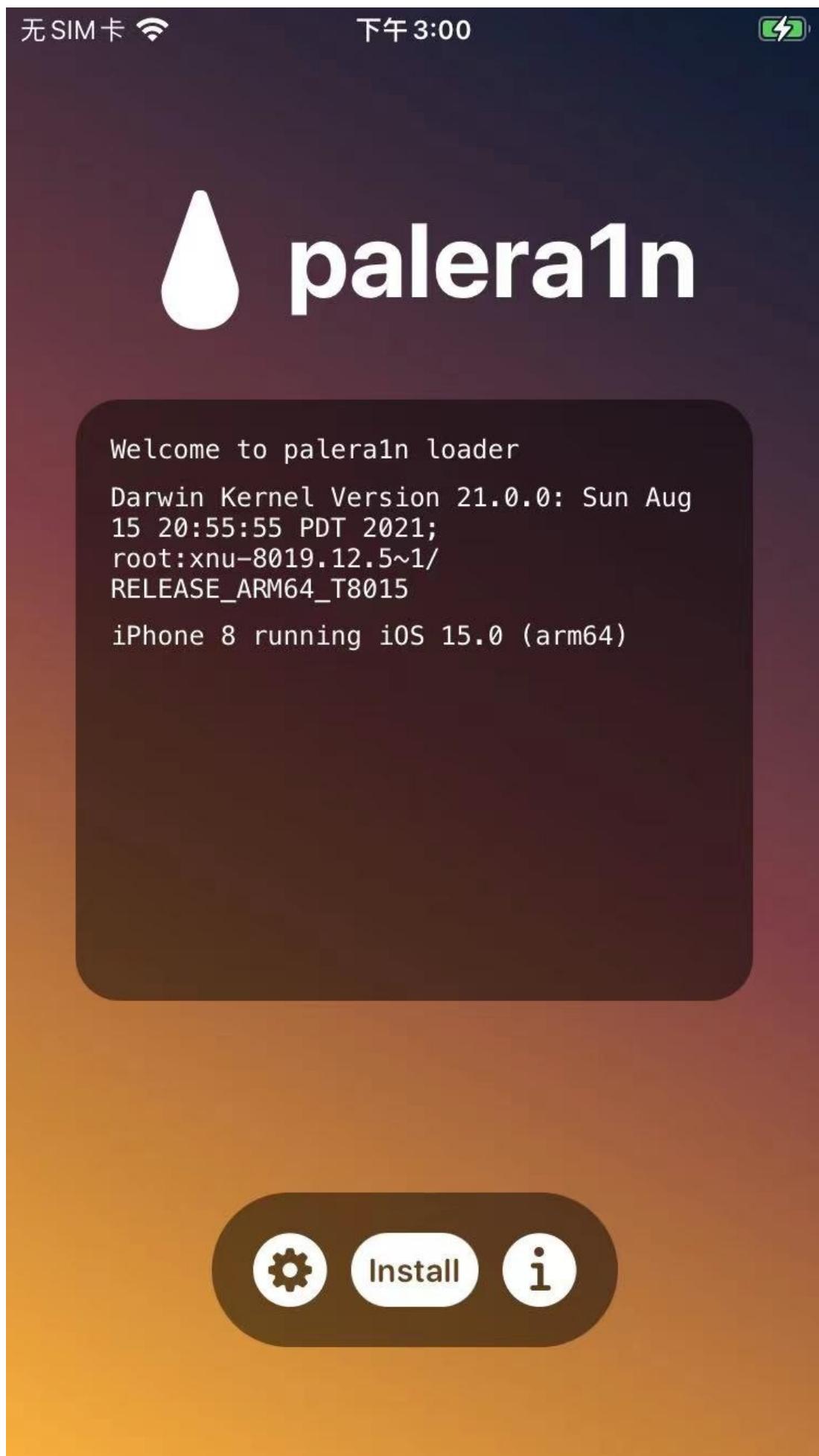
## palera1n loader app

- palera1n的app = palera1n loader = palera1n loader app

打开palera1n的app后：

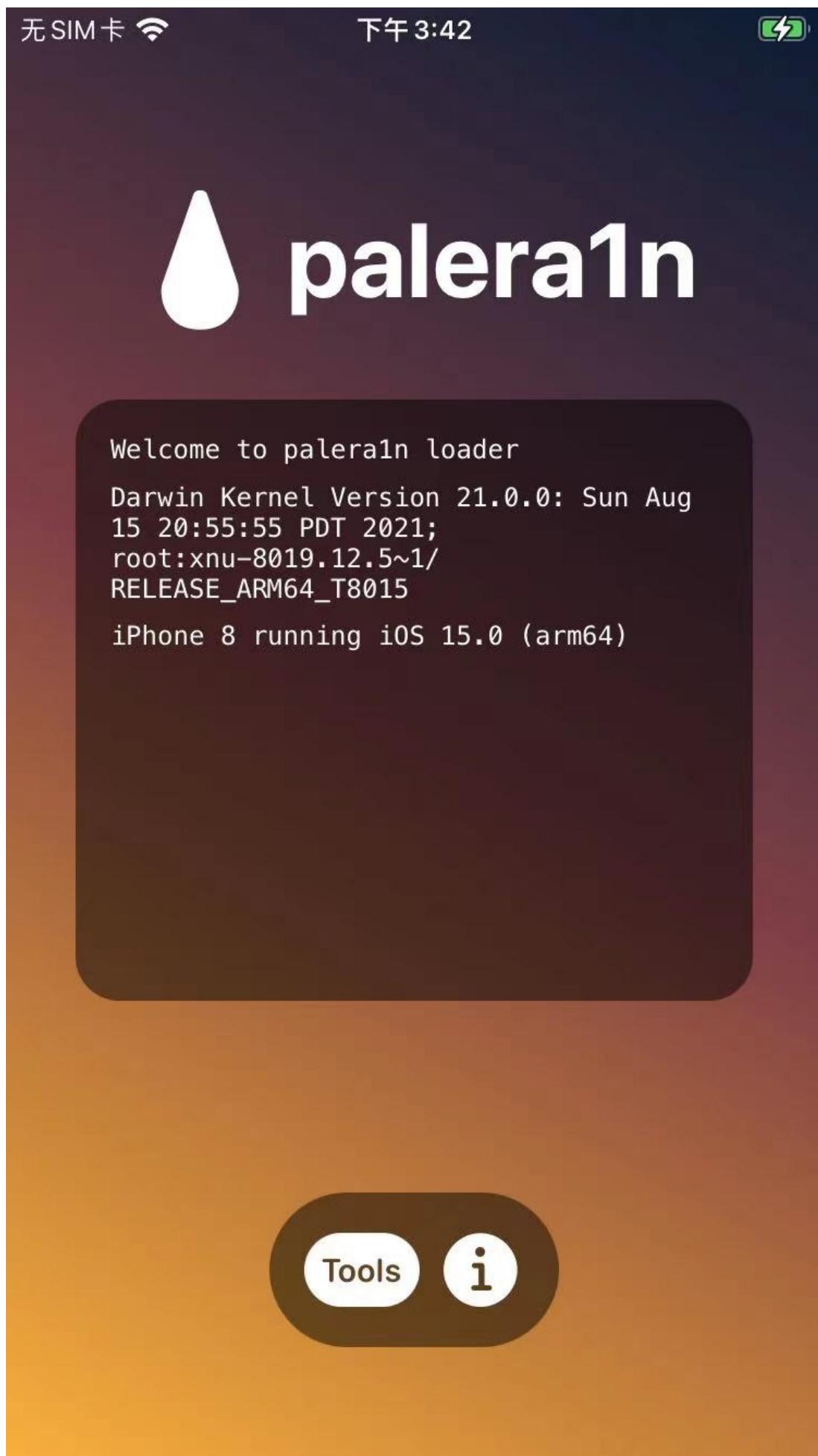
### 首页

- 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：

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





# XinaA15

TODO:

整理出，用XinaA15给iOS 15.1的iPhone11越狱的过程

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：2023-03-02 22:18:48

## 越狱后

- 【整理】已越狱后的iPhone和iOS相关知识
- 【整理】已越狱iOS的ipa安装工具：Cydia Impactor
- 【整理】已越狱的iPhone有哪些有用的有价值的扩展插件
- 【已解决】iPhone6中重新激活和开启越狱状态
- 【记录】新iPhone测试机iPhone7P越狱环境准备

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：2022-06-21 13:57:34

## palera1n越狱后

### 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

**procurus-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。

## 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
...
```

## 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](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](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](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/unrar\\_6.1.4\\_iphoneos-arm.deb](https://apt.procurs.us/pool/main/iphoneos-arm64/1700/unrar/unrar_6.1.4_iphoneos-arm.deb)
  - p7zip
    - [https://repo.chimera.sh/debs/p7zip\\_16.02\\_iphoneos-arm.deb](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](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>

## palera1n越狱后，重启丢失越狱，如何恢复越狱

```
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

## 恢复越狱

TODO:

- 【记录】给之前用checkra1n越狱的iPhone6恢复越狱
- 

现在主流越狱工具，比如 `unc0ver`，都是 半完美越狱：iOS（iPhone）重启后，越狱就丢失了

-> 具体现象是：点击 Cydia 等软件会闪崩无法打开

此时，就需要去：恢复越狱

即把之前越狱的流程再走一遍

-> 即可恢复越狱状态。

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：2022-10-27 21:37:40

# 文件管理

越狱后的iPhone，可以有很多文件管理方面的工具：

- 文件管理
    - ssh
      - OpenSSH
      - 免密登录
      - scp：导入 导出 文件
    - Filza
      - 文件管理
    - 爱思助手
  - 安装ipa
    - Filza
    - 爱思助手
- 

TODO：

- 【已解决】从已越狱iPhone中拷贝文件到Mac中
- 【已解决】已越狱iOS中通过Cydia安装文件管理器
- 【已解决】从已越狱iPhone中拷贝文件到Mac中

其他：

- iFile（收费）
  - 【未解决】iPhone中安装和使用iFile查看iOS是否已越狱
  - 【未解决】Cydia安装BigBoss源的iFile出错：无法购买Cydia is not yet prepared to accept money

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：2022-10-23 14:53:35

## 包管理器

- 【整理】越狱iOS包管理器越狱版AppStore

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: 2022-06-21 14:06:11

# Cydia

- Cydia
  - 【整理】越狱iOS包管理器越狱版AppStore：cydia
  - 【已解决】iPhone6中用checkra1n安装Cydia
  - 【记录】iPhone6中试用Cydia
  - 【记录】已越狱的iOS中的Cydia的情况：已安装源和插件
  - 【已解决】已越狱iPhone中Cydia中安装Cydia Substrate
  - 【未解决】Cydia安装BigBoss源的iFile出错：无法购买Cydia is not yet prepared to accept money

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：2022-06-21 14:06:23

## 插件

其他一些插件

- SSL Kill Switch 2
  - 【已解决】越狱iPhone中安装越狱插件：SSL Kill Switch 2
- 终端
  - 【已解决】越狱iPhone中安装Cydia插件：终端工具
    - Mterminal
    - NewTerm 2
    - TODO: 抽空去试试

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: 2022-06-21 14:01:56

## Apple File Conuit "2"

简称： AFC2

- 【已解决】已越狱iPhone用Cydia安装Apple File Conuit "2"
- 【已解决】Cydia安装AFC2报错：Can't find a source to download version 1.1.1of apt.zscool.net.arm64:iphoneos-arm

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：2022-06-21 14:00:35

## AppSync Unified

TODO:

- 【已解决】已越狱iPhone中用插件AppSync
  - 【记录】越狱iPhone7P通过Cydia安装插件：AppSync Unified
  - 【已解决】已越狱iPhone中用插件AppSync
- 

- AppSync = AppSync Unified
  - 用途：让系统不再验证签名，以免安装应用失败

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：2022-10-23 14:52:48

## OpenSSH

- 【记录】iPhone中用Cydia安装SSH插件：OpenSSH

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：2022-06-21 14:01:00

## Filza

- 安装64位的Filza
  - 概述
    - Cydia源: <http://tigisoftware.com/cydia/> -> TIGI Software -> 全部工具 -> Filza File Manager 64-bit
  - 详解
    - 【记录】越狱iPhone6P中用Cydia安装Filza File Manager 64-bit
- 【记录】已越狱iPhone中使用Filza File Manager
- 【已解决】越狱iOS中用Cydia安装Filza File Manager
- 【记录】越狱iPhone7P中安装Filza
- 【记录】越狱iPhone7P中安装64位的Filza File Manager 64-bit
- 【记录】iPhone7P中用Filza安装抖音ipa

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: 2022-06-21 14:02:58

## iCleaner Pro

- 【已解决】Cydia中如何临时的禁止插件生效而不是只能卸载掉

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: 2022-06-21 14:02:05

## Mterminal

- 【已解决】越狱iPhone中用Cydia安装终端工具：Mterminal

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: 2022-06-21 14:01:31

## 使用心得

插件安装心得：有时候新增源后里面是空的，但是其实可以搜索到需要的插件的

比如之前新增了：

- 源：`Ivano Bilenchi's Repo`
- 地址：`https://ib-soft.net/cydia` 但是进入后，全部是空的 没有想要安装的插件：iCleaner Pro 但此时，去搜索，是可以搜到：`iCleaner Pro` 的，即可正常继续安装iCleaner Pro了。

详见：【记录】iPhone X安装Cydia插件：iCleaner Pro

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：2022-06-21 13:42:08

# Sileo

- 【整理】越狱iOS包管理器越狱版AppStore: Sileo

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: 2022-06-21 14:06:40

## 爱思助手

- 【记录】Mac中用爱思助手检测和确认iPhone是否已越狱
- 【记录】已越狱的iOS中用爱思助手安装app软件：微信
- 【已解决】用Mac版爱思助手给iOS13的iPhone7越狱
- 【记录】iPhone6中试用爱思极速版
- 【已解决】Mac中爱思助手看不到iPhone的越狱文件系统全部文件内容
- 【已解决】Mac中用爱思助手安装抖音ipa到越狱iPhone中

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: 2022-06-21 14:06:53

## 安装ipa

- Filza
  - 【整理】iOS逆向心得：越狱iPhone异常时的现象：Filza安装ipa卡死

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：2022-06-21 13:40:13

## 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: 2022-06-19 08:09:37

## 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】iOS的半越狱和全越狱
- 【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
- 

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：2023-03-09 23:11:17