

---

# 目录

前言	1.1
安卓root概览	1.2
安卓模拟器root	1.3
安卓手机root	1.4
root流程	1.4.1
BL解锁	1.4.1.1
fastboot mode	1.4.1.2
root相关工具	1.4.2
TWRP	1.4.2.1
Magisk	1.4.2.2
给Android13的Pixel5去root	1.4.3
下载安装最新版Magisk	1.4.3.1
解锁Bootloader	1.4.3.2
找到要打补丁的boot.img文件	1.4.3.3
用Magisk去给boot.img打patch	1.4.3.4
用Magisk写入patch后的boot.img	1.4.3.5
Magisk中root相关设置	1.4.3.6
给LineageOS19的的小米8去root	1.4.4
下载安装Kitsune Mask	1.4.4.1
解锁Bootloader	1.4.4.2
找到要打补丁的boot.img文件	1.4.4.3
用Kitsune Mask去给boot.img打patch	1.4.4.4
用Kitsune Mask写入patch后的boot.img	1.4.4.5
弹框：需要修复运行环境	1.4.4.5.1
Kitsune Mask中root相关设置	1.4.4.6
root相关	1.5
A/B槽位	1.5.1
OPPO R11s	1.5.2
root心得	1.6
附录	1.7
参考资料	1.7.1

# Android逆向：开启root

- 最新版本： `v2.0.0`
- 更新时间： `20241114`

## 简介

总结安卓逆向期间涉及的给安卓root。先是概览；然后是分别介绍安卓模拟器和安卓真机的root；之后详细介绍安卓真机的root的流程，包括Bootloader解锁、fastboot mode等，和涉及到的工具：TWRP、Magisk等；并且用实例去介绍如何给Android13的Pixel5去root，包括下载安装最新版Magisk、解锁Bootloader、用Magisk去给boot.img打patch、用Magisk写入patch后的boot.img和Magisk中root相关设置。以及相关知识和设备：A/B槽位、OPPO R11s的root；以及整理root相关心得。

## 源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

### HonKit源码

- [crifan/android\\_re\\_enable\\_root: Android逆向：开启root](#)

### 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

- [Android逆向：开启root book.crifan.org](#)
- [Android逆向：开启root crifan.github.io](#)

### 离线下载阅读

- [Android逆向：开启root PDF](#)
- [Android逆向：开启root ePub](#)
- [Android逆向：开启root 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：2024-11-14 11:09:39

# 安卓root概览

Android逆向期间，往往前提是需要：一个root的安卓手机。

就会涉及到，如何给安卓手机root。

此处介绍Android的root的相关内容。

## 什么是安卓的root?

- 背景：Android系统（底层是Linux系统）的最高权限的用户叫做：root 用户
- 安卓的root=获取最高权限=获取root用户的权限=获取超级用户的权限
  - 类似于：iOS系统中的越狱 =给iPhone 越狱
  - 有了root权限后：就可以做很多（普通的默认的用户权限无法做的）事情 = 可以访问和修改你手机几乎所有的文件

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：2023-08-25 22:13:10

## 安卓模拟器root

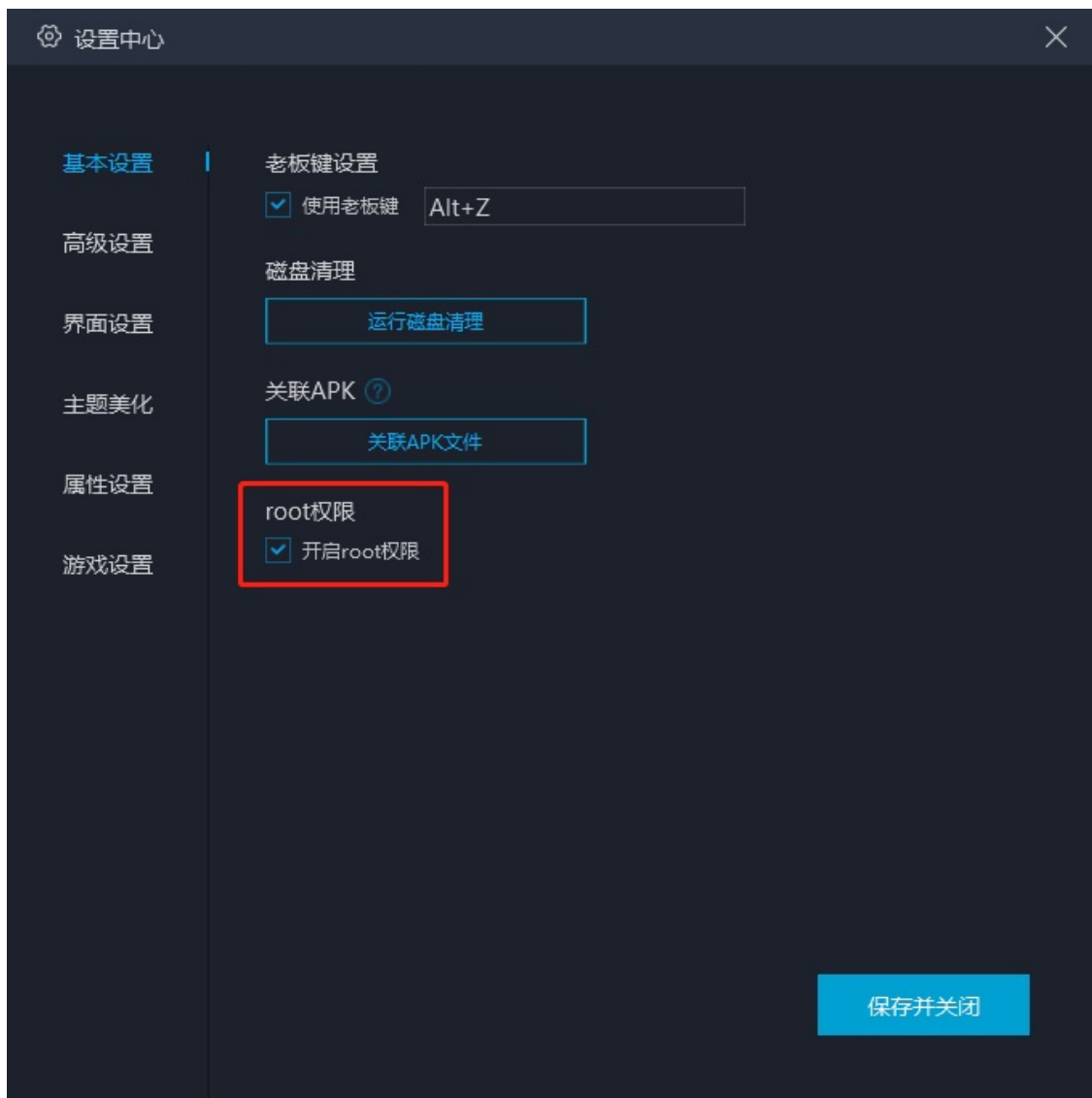
Android逆向期间，如果要逆向的app，可以在安卓模拟器中正常安装和运行，那么，其实使用模拟器去折腾，也是一个比较好的选择，因为：

安卓的root权限，对于多数安卓模拟器来说，都能很方便的支持，毕竟基本上就是一个参数的开启的事情。

目前已知的，相对还算好用的安卓模拟器，且支持root的有：

- 夜神Nox

- 
- 网易Mumu
  - <https://mumu.163.com/>



# 安卓手机root

给安卓手机=安卓真机，去root：

- 早期：是个简单的事情
- 现在：往往，是个复杂、麻烦的事情

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：2022-10-30 17:36:56

# root流程

TODO:

- 【已解决】给Android 11的Google Pixel3去开启root权限
- 【未解决】如何给OPPO R11s开通root权限
- 【记录】root安卓手机OPPO R11s的root环境初始化

---

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: 2022-10-30 17:56:52



# BL解锁

TODO:

- 【未解决】如何给OPPO R11s去Bootloader解锁
- 

- BL解锁 = Bootloader解锁

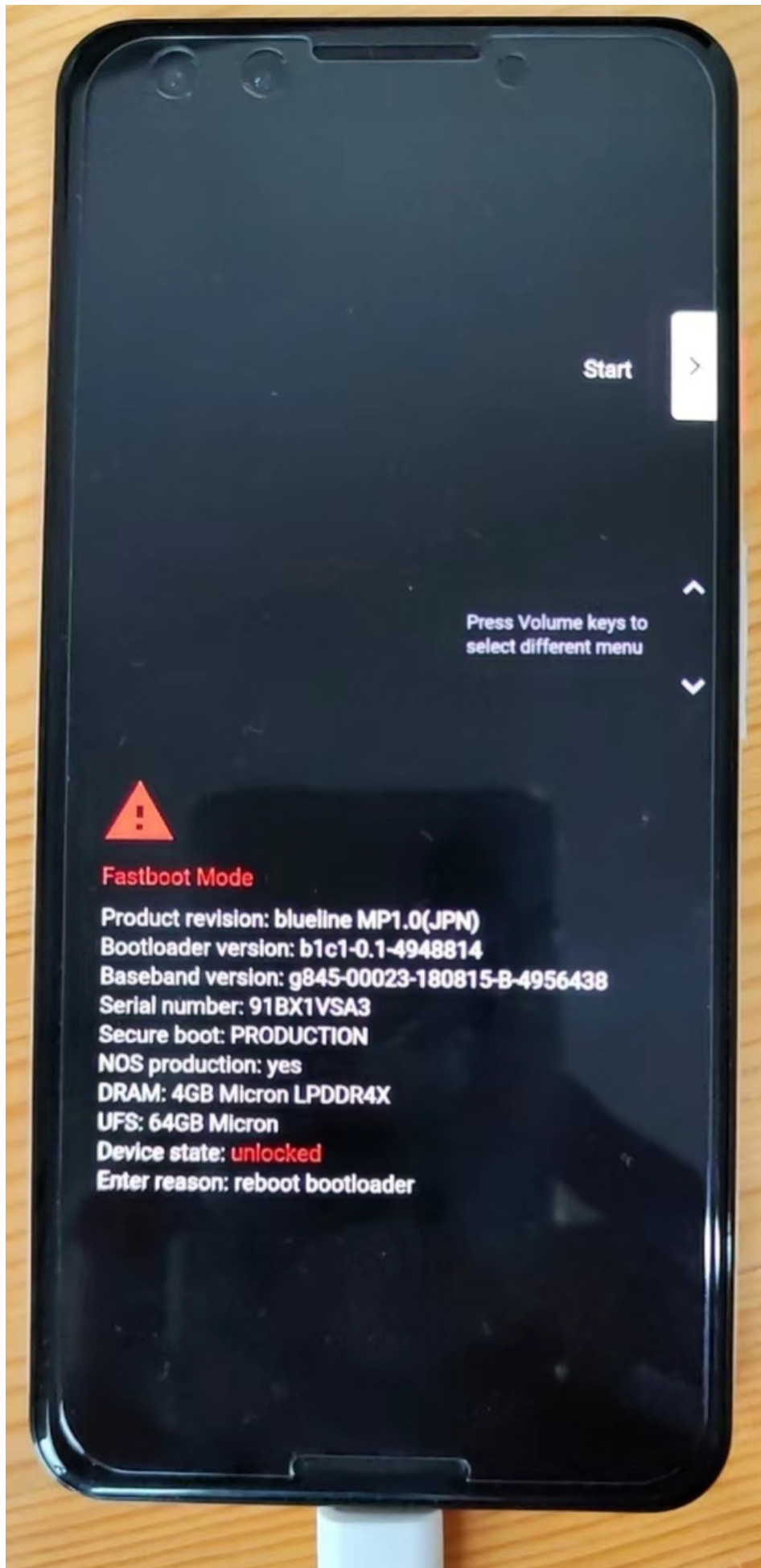
crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2022-10-30 17:56:34

# fastboot mode

TODO:

- 【已解决】 安卓手机什么是fastboot mode
- 【未解决】 Android 8.1的OPPO R11s无法进入bootloader的fastboot mode

- 
- fastboot mode
    - = 刷机模式 = bootloader mode = download mode = 下载模式
    - 长什么样
      - Google Pixel 3
      -





## root工具

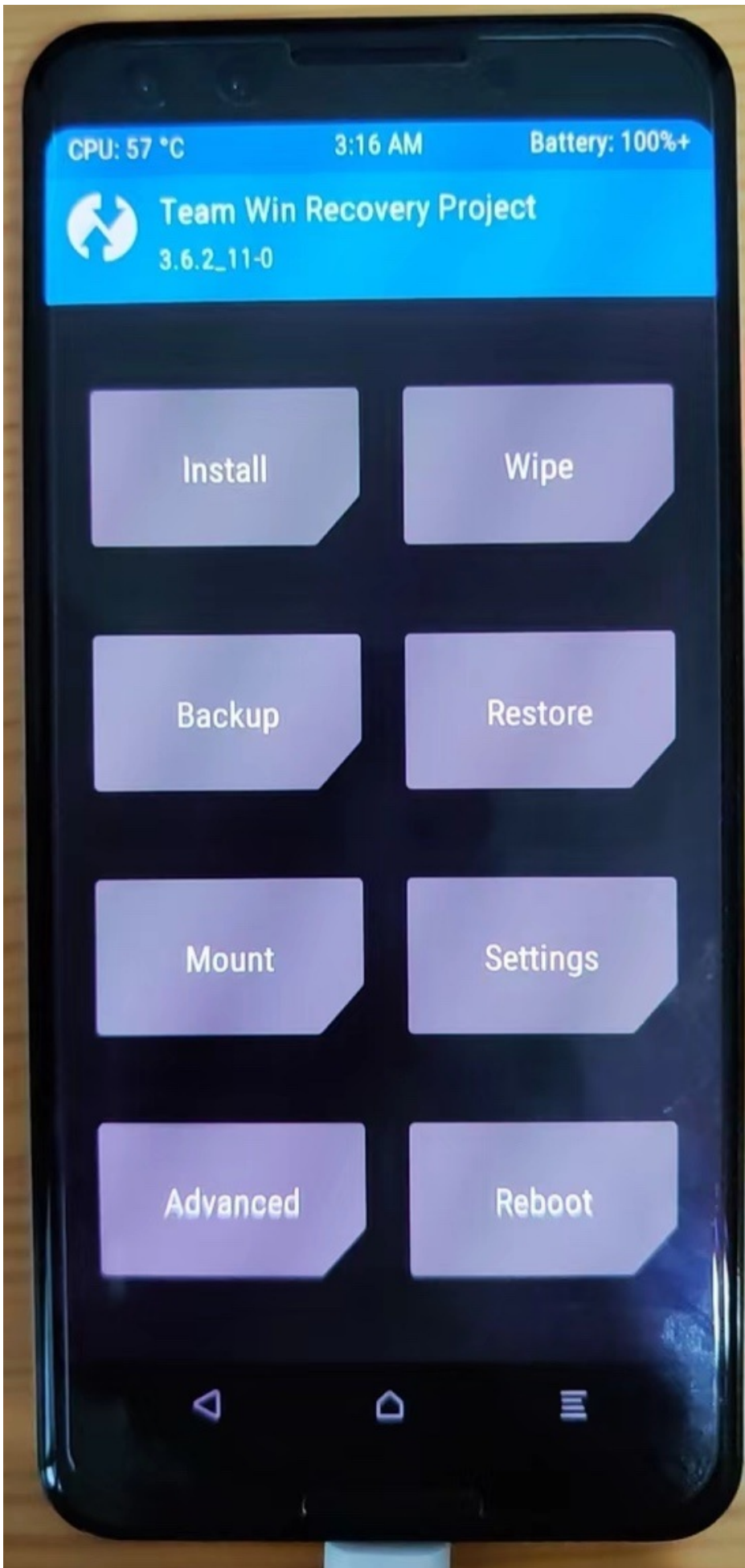
crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2022-10-30 15:40:12

# TWRP

TODO:

- 【已解决】给Android 11的Google Pixel3去刷第三方Recovery: TWRP
- 【已解决】Google Pixel3如何进入TWRP
- 【记录】Google Pixel3中的TWRP界面和功能
- 【未解决】给OPPO R11s开启root权限: 用TWRP的Recovery刷Magisk
- 【未解决】给OPPO R11s刷第三方Recovery: TWRP
- 【记录】OPPO R11s重启后进入奇兔刷机Recovery模式TWRP

- 
- TWRP
    -







# Magisk

TODO:

- 【未解决】给OPPO R11s开启root权限：用非TWRP的Recovery模式安装刷入Magisk
- 【总结】Magisk Manager使用心得：root超级用户权限管理
- 【整理】安卓root工具：Magisk
- 【已解决】OPPO R11s中安装Magisk并给boot.img打补丁
- 【已解决】Magisk中安装中去Patch启动镜像boot.img
- 【记录】已root的Google Pixel3中的Magisk相关信息
- 【记录】OPPO R11s中Magisk Manager初始化和配置参数
- 【记录】root安卓手机OPPO R11s中升级Magisk Manager
- 【已解决】OPPO R11s中Magisk Manager升级后提示：不支持的Magisk版本
- 【整理】Google Pixel3中的Magisk Manager使用心得
- 【记录】手动下载和升级Magisk到最新版本
- 【记录】Magisk Manager版本升级
- 【未解决】Magisk Manager模块从本地安装无法识别选择apk文件

- 
- Magisk
    - 新版： v21
      -

11:49     34%

主页  

仅从官方 GitHub 页面下载 Magisk。未知来源的文件可能具有恶意行为! 不再显示

 **Magisk** 更新

当前 **21.0 (21000)**    A/B **是**  
Ramdisk **是**                    SAR **是**

 **App** 更新

最新 **25.2 (25200) (33)**  
当前 **23.0 (23000)**  
包名 **com.topjohnwu.magisk**

 **测试 SafetyNet 证明**

 **卸载 Magisk**

### 支持开发

Magisk 将一直保持免费且开源，向开发者捐赠以表示支持。

@topjohnwu

@diareus  

- 最新版: v25





## 新版Magisk中没有模块的在线搜索了

TODO:

- 【未解决】新版Magisk中没有在线搜索安装模块了

## Zygisk

TODO:

- 【已解决】Magisk中的: Zygisk

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](https://creativecommons.org/licenses/by/4.0/)发布 all right reserved, powered by Gitbook最后更新: 2022-10-30 17:52:39

## 给Android13的Pixel5去root

对于 `Android 13` 的安卓手机 `Google Pixel 5` , 此处去用 `Magisk` 去root。

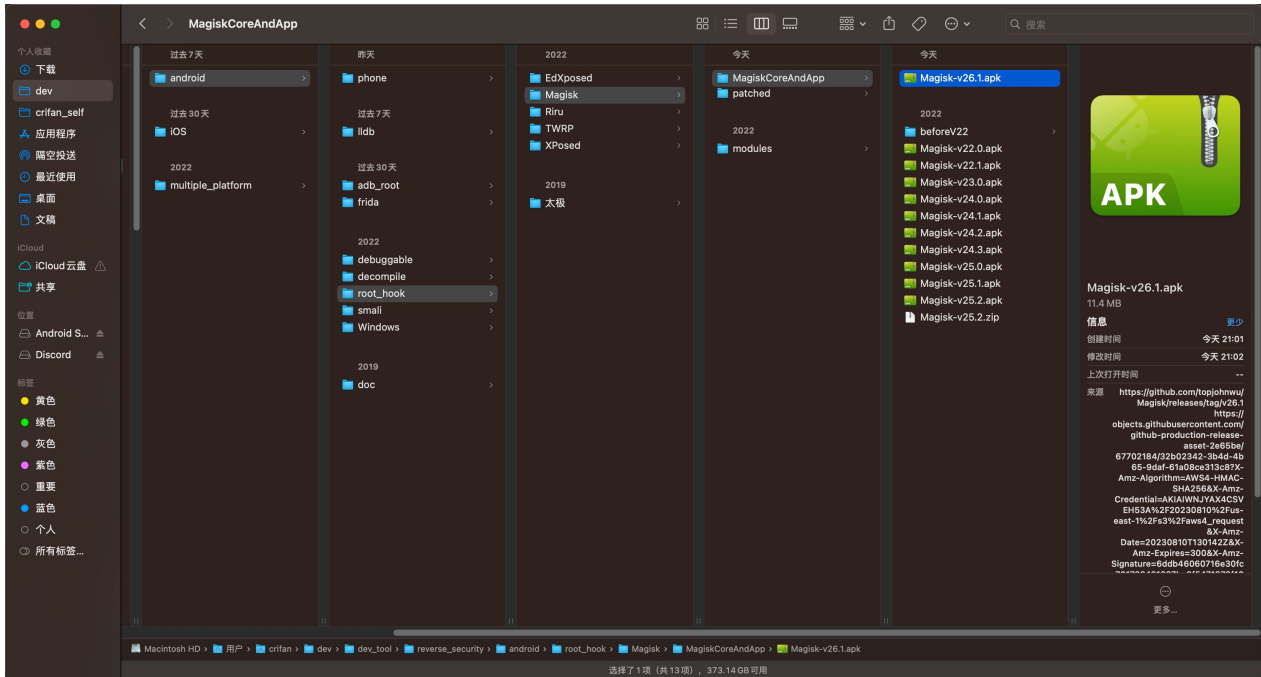
crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2023-08-16 22:06:22

## 下载和安装最新版Magisk

此处最新版是 `v26.1`，所以去 [Release Magisk v26.1 · topjohnwu/Magisk \(github.com\)](https://github.com/topjohnwu/Magisk/releases/tag/v26.1)，找到：

<https://github.com/topjohnwu/Magisk/releases/download/v26.1/Magisk-v26.1.apk>

并下载，得到：`Magisk-v26.1.apk`



然后下载到安卓手机Pixel5中：

```
adb push Magisk-v26.1.apk /sdcard/Download/
```

再去用文件管理器 文件极客 去安装：点击下载目录中的 `Magisk-v26.1.apk`：

21:05     

← 已下载的文件  

全部    Download

今天



Magisk-v26.1.apk  
11.41 MB, 2分钟前



weixin8040android2420\_arm64\_1....  
261 MB, 3小时前



正在安装应用...

× 停止



去安装，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: 2023-08-16 22:25:55

# 解锁Bootloader

先确保

- 系统 设置 -> 开发者选项 -> 设置 -> 已开启：OEM解锁
  -

## OEM 解锁

允许解锁引导加载程序



## 正在运行的服务

查看和控制当前正在运行的服务

## WebView 实现

Android System WebView

## 系统自动更新

设备重启时执行更新



## DSU Loader

Load a Dynamic System Update Image

## 系统界面演示模式

## 快捷设置开发者图块

调试

## USB 调试

连接 USB 后启用调试模式



## 撤消 USB 调试授权

## 无线调试

连接到 WLAN 后启用调试模式



## 停用 adb 授权超时功能

停用以下功能：如果系统在默认时间（7 天）或用户配置的时间（最短 1 天）内未重新建立连接，就自动撤消 adb 授权。

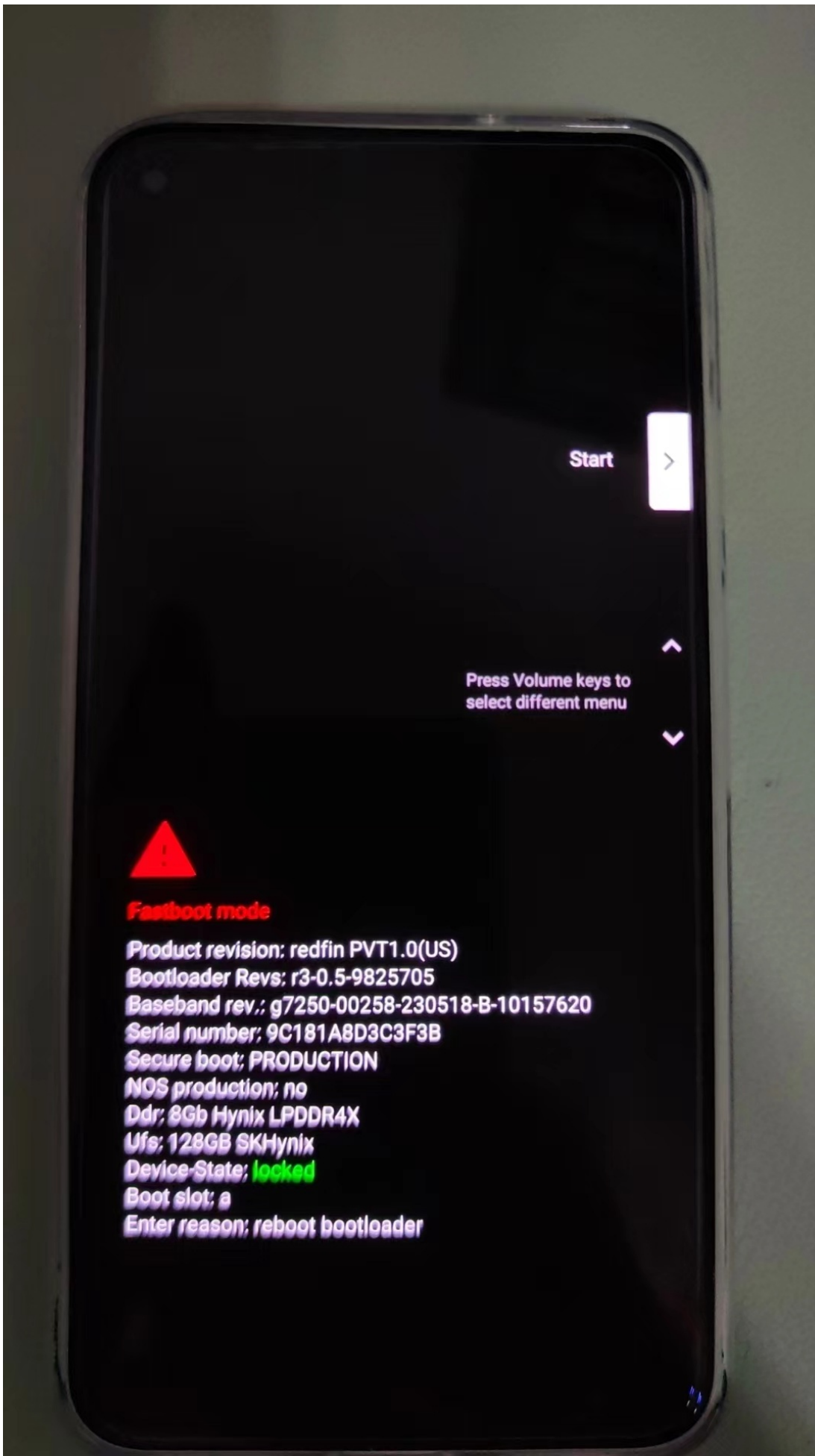


快速设置开发者图块

再去解锁 Bootloader ：

```
adb reboot bootloader
```

此时安卓手机会进入 Fastboot模式 ， 其中能看到 Device-State: locked ：



表示Bootloader未解锁

再去：

```
fastboot devices
```

确保能看到，处于的安卓手机设备：

比如：

```
→ MagiskCoreAndApp fastboot devices  
9C181A8D3C3F3B fastboot
```

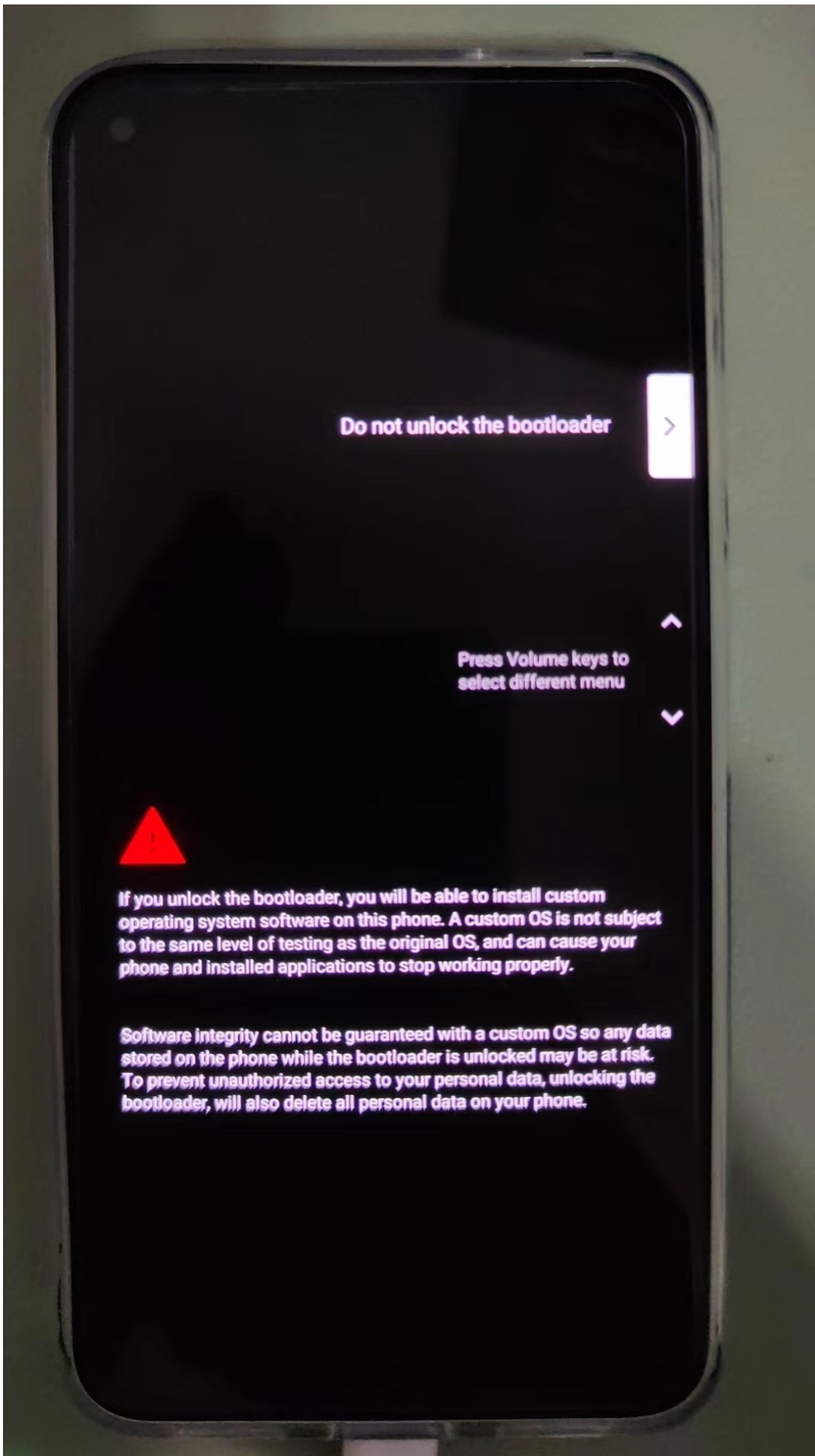
- 如果想要查看详情，可以加参数 `-l`

```
→ MagiskCoreAndApp fastboot devices -l  
9C181A8D3C3F3B fastboot usb:1048576X
```

再去用：

```
fastboot flashing unlock
```

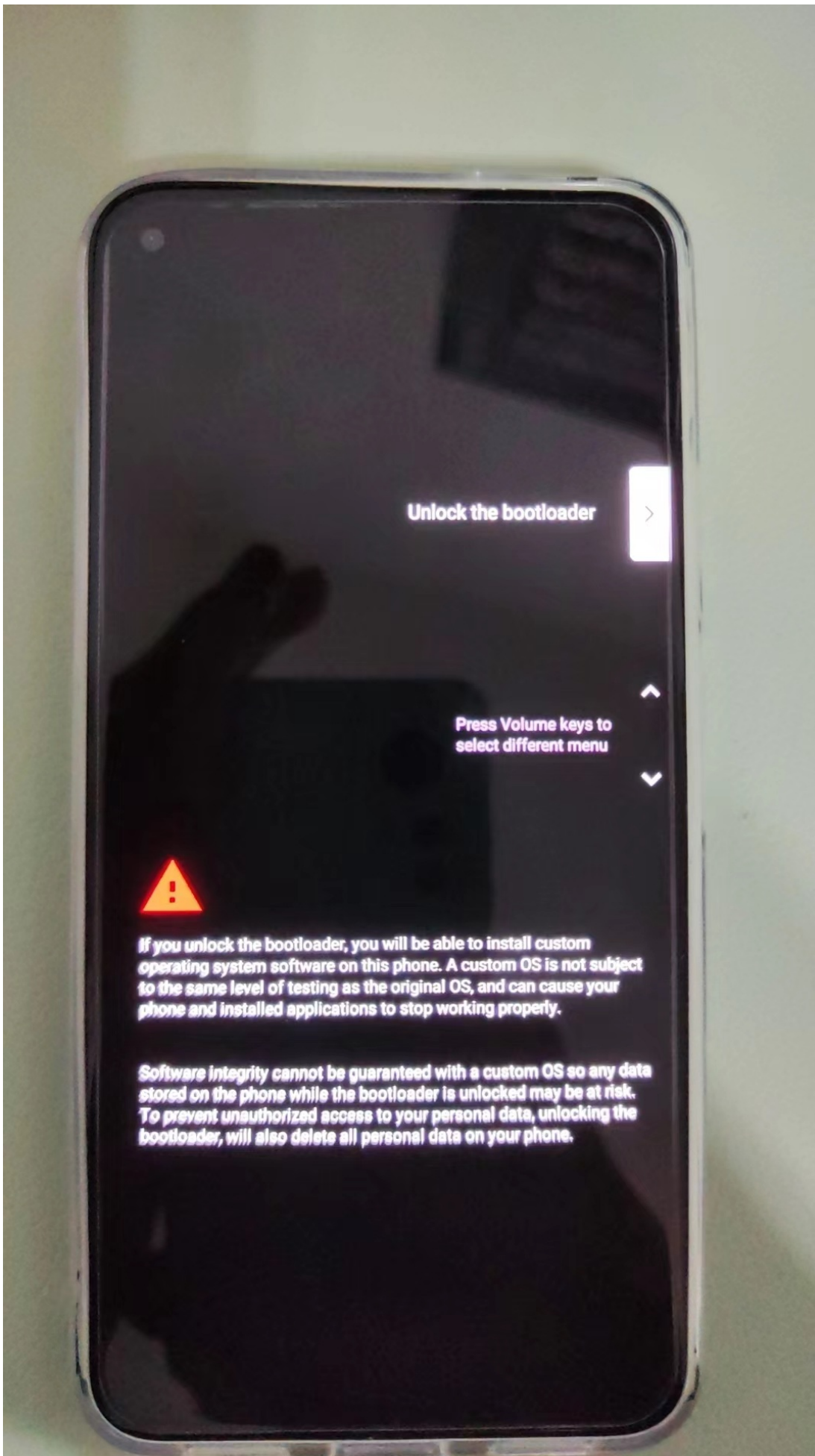
手机会进入Bootloader解锁页面：



按音量键加减，切换到：

Unlock the bootloader





Unlock the bootloader



Press Volume keys to  
select different menu



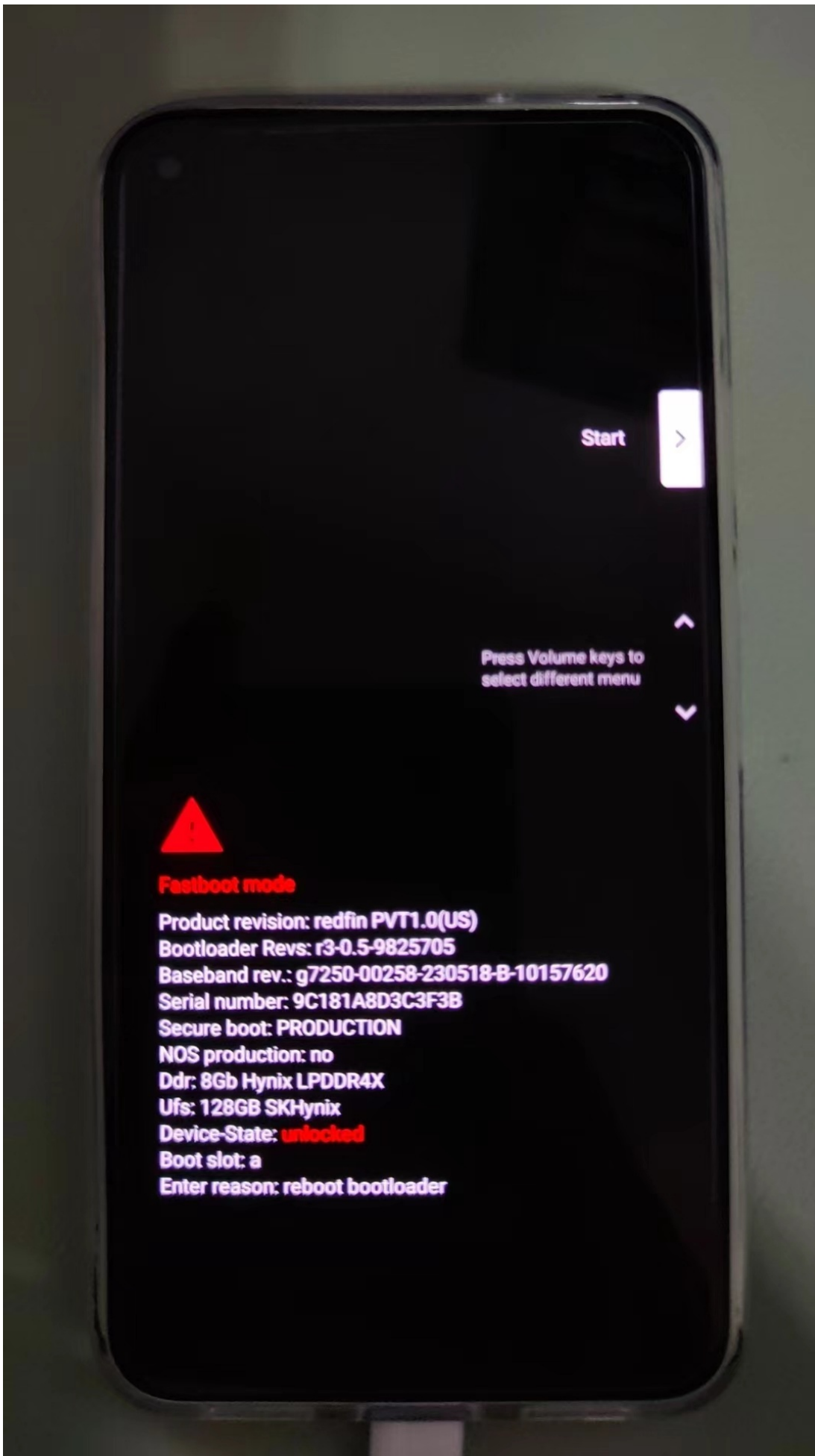
**If you unlock the bootloader, you will be able to install custom operating system software on this phone. A custom OS is not subject to the same level of testing as the original OS, and can cause your phone and installed applications to stop working properly.**

**Software integrity cannot be guaranteed with a custom OS so any data stored on the phone while the bootloader is unlocked may be at risk. To prevent unauthorized access to your personal data, unlocking the bootloader, will also delete all personal data on your phone.**

再按 电源键 表示确认

会去重启安卓手机Pixel5, 重启后再次进入了 Fastboot Mode

会看到 Device-State: unlocked



表示：Bootloader已解锁

然后：

```
fastboot reboot
```

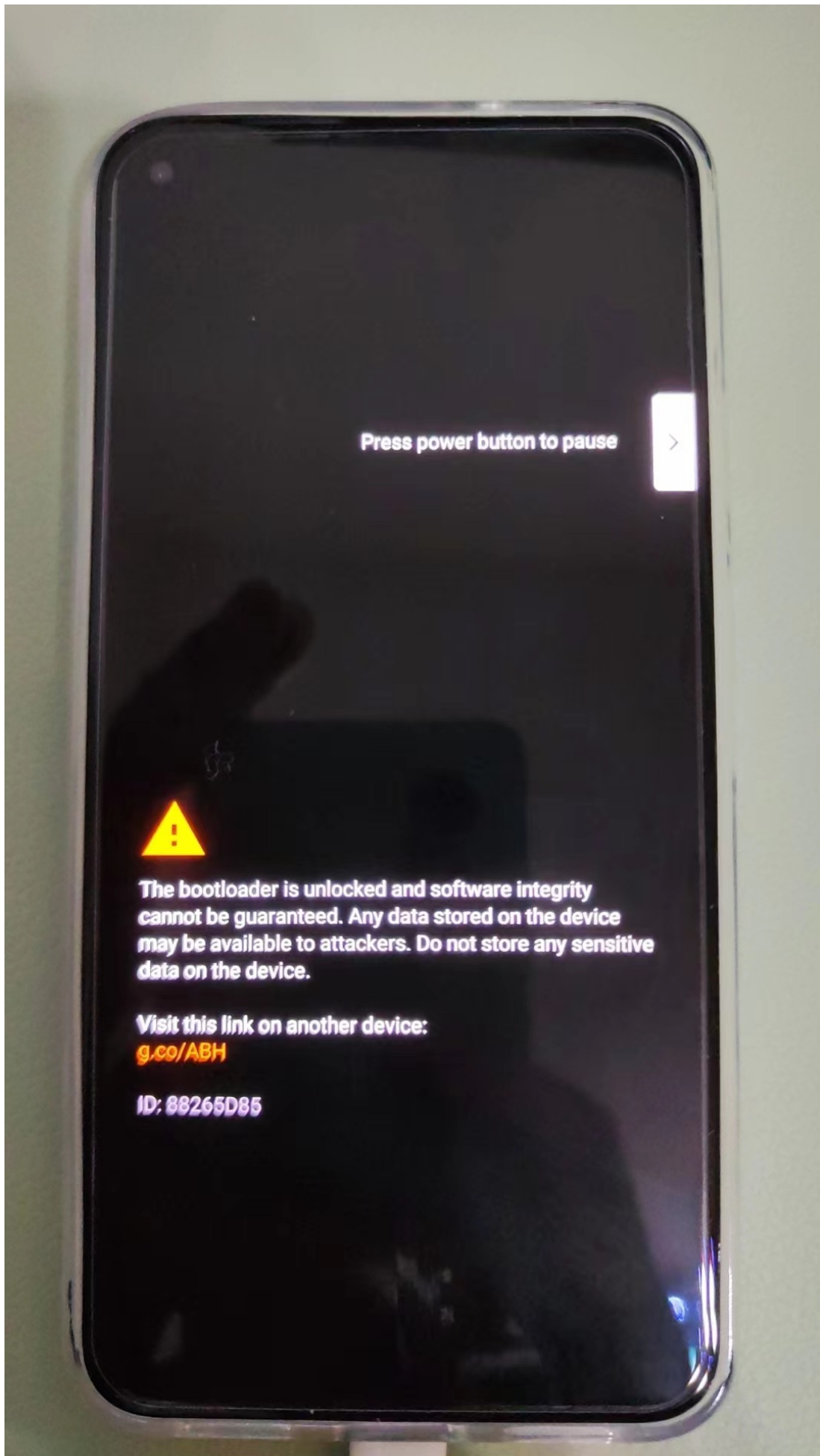
会重启安卓手机

注：重启后

- 此时就相当于一个新的手机
  - 就像之前重刷了官网的Android13的ROM一样
    - -> 需要一步步初始化设置，直到进入系统桌面
  - 另外最好重新去：`开启USB调试`
    - 供后续开发调试用
- `系统 -> 设置 -> 开发者选项 -> 设置`
  - `OEM解锁`：已经变成灰色了=不可勾选
    -



- -> 表示 已解锁 Bootloader了，无法再次点击开启
- 对于已解锁Bootloader的安卓手机（此处的Pixel5），后续每次重启时，都会有相关的提示 `The bootloader is unlocked`  
...
  -







## 找到要打补丁的boot.img文件

对于后续Magisk的root来说，需要找到：要打补丁的 `boot.img` 文件：

先去从Android官网，下载适配当前安卓手机 Google 的 Pixel5 的镜像image：

- Nexus 和 Pixel 设备的出厂映像 | Google Play services | Google for Developers
  - <https://developers.google.cn/android/images>

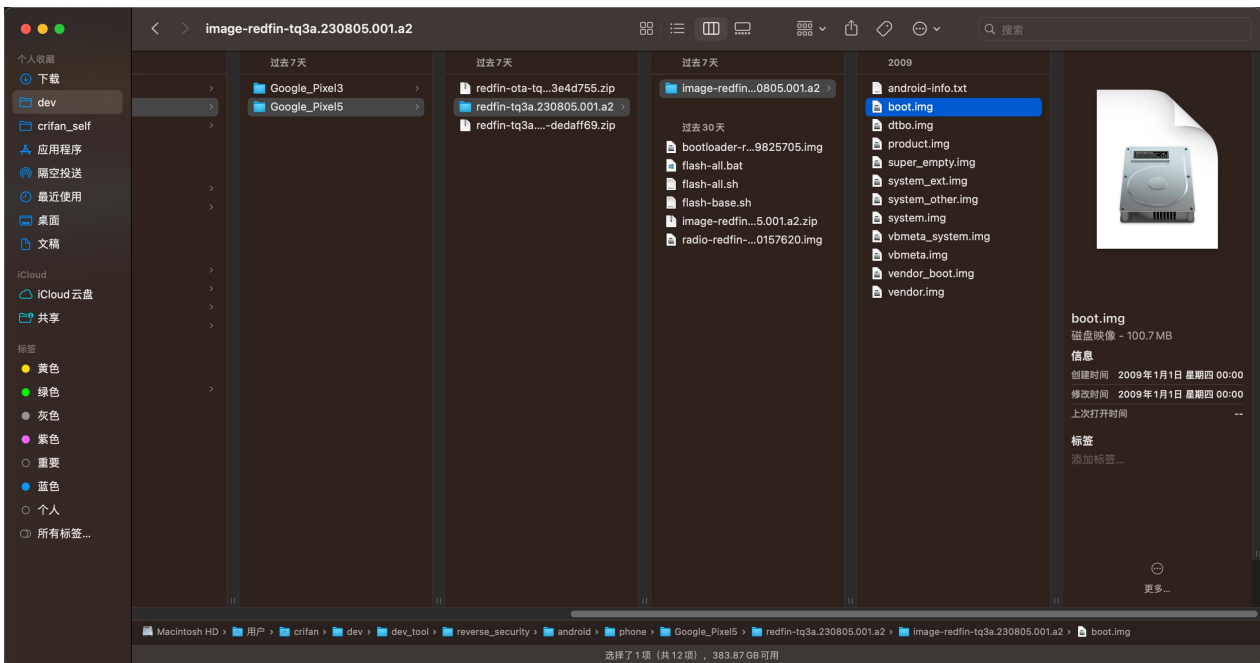
->

<https://dl.google.com/dl/android/aosp/redfin-tq3a.230805.001.a2-factory-dedaff69.zip?hl=zh-cn>

下载得到：`redfin-tq3a.230805.001.a2-factory-dedaff69.zip`

解压得到文件夹：`redfin-tq3a.230805.001.a2`

找到：`redfin-tq3a.230805.001.a2/image-redfin-tq3a.230805.001.a2` 中的 (96MB的) `boot.img`



就是我们后续要打补丁的文件了。

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：2024-11-14 09:44:39

## 用Magisk去给boot.img打patch

将 boot.img 下载到Pixel5手机中（的下载目录）：

```
adb push boot.img /sdcard/Download
```

再去Magisk中：

- 概述： Magisk -> 安装 -> 选择并修补一个文件 -> （从下载目录中）选择 boot.img -> 开始 -> 完成 ->输出log中会有打了patch的 boot.img
- 详解：
  - 点击打开Magisk
    -

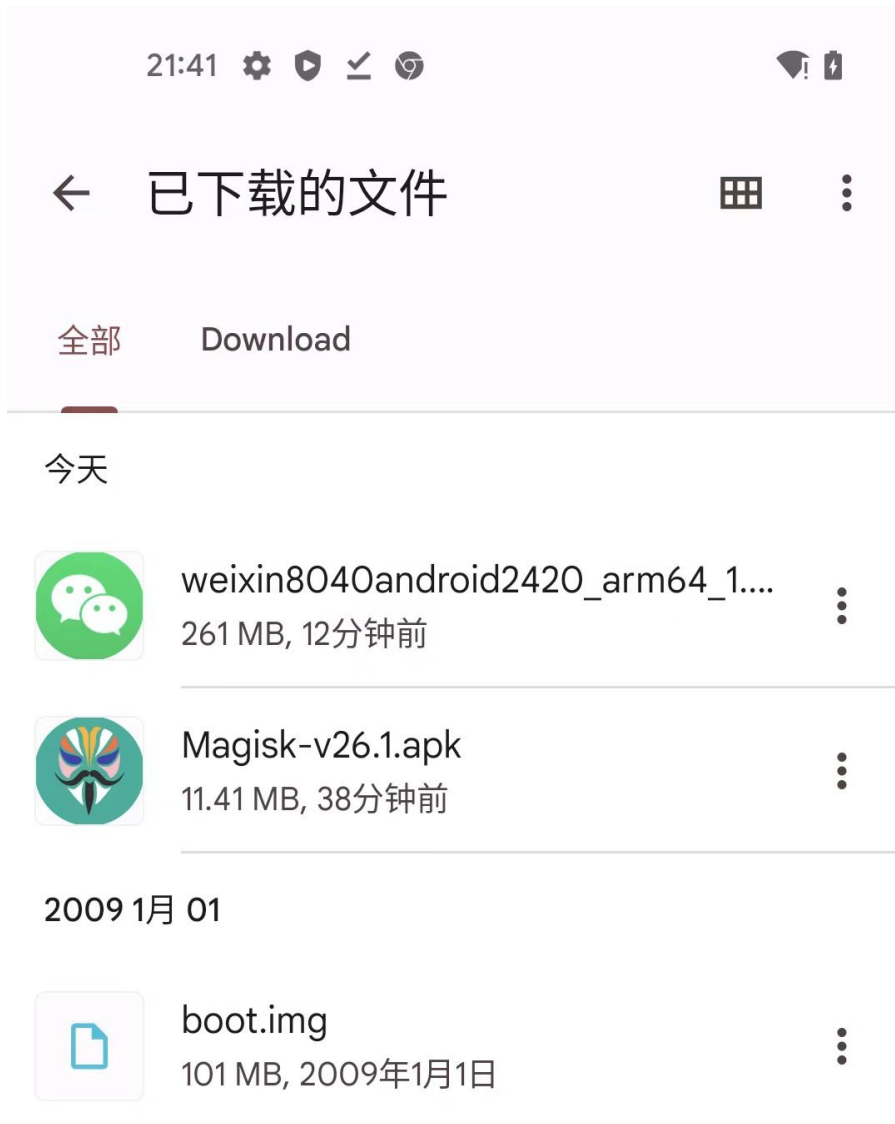


- 点击Magisk中的主页中的：安装





- 选择并修补一个文件
- (从下载目录中) 选择 `boot.img`
  -



已安装Magisk。

打开

- 开始
- 完成
-





- 输出log中会有打了patch的 boot.img

```
Output file is written to
/storage/emulated/0/Download/magisk_patched-26100_bMrsR.img
```

- 其中:
  - /storage/emulated/0/Download/magisk\_patched-26100\_bMrsR.img
    - == /sdcard/Download/magisk\_patched-26100\_bMrsR.img
    - 就是我们要的, 打好了 patch 的 boot.img
- 把打了patch的 boot.img 上传到电脑端, 供后续使用
  - 把此处打了patch的 boot.img 是 magisk\_patched-26100\_bMrsR.img , 从安卓手机中上传到电脑 (Mac) 中

```
adb pull /sdcard/Download/magisk_patched-26100_bMrsR.img .
```

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: 2024-11-14 09:48:08

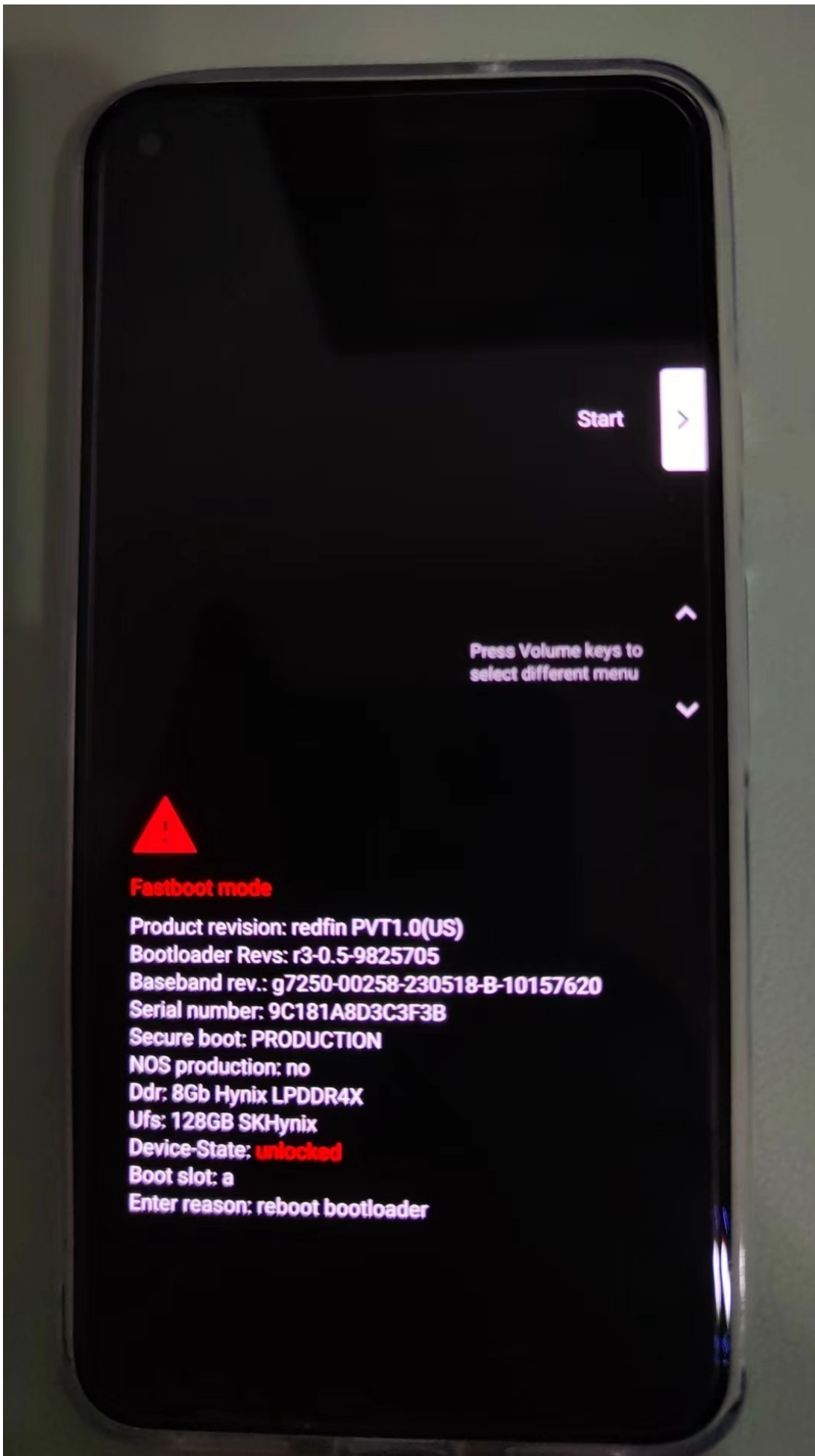
## 用Magisk写入patch后的boot.img

**先临时写入patch后的boot.img，启动系统，使得Magisk有了root权限**

(1) 先：进入Fastboot模式：

```
adb reboot bootloader
```

重启后，手机进入Fastboot Mode:



注:

- 最好再用命令 `fastboot devices` 确认的确进入了Fastboot mode == 可以找到Fastboot的设备

```
→ GooglePixel5 fastboot devices
9C181A8D3C3F3B fastboot
```

(2) 再: (此处是临时) 写入用Magisk打了patch的boot.img

```
fastboot boot magisk_patched-26100_tpJtT.img
```

• 说明

- `magisk_patched-26100_tpJtT.img` 是之前用Magisk 打了patch后的boot.img
- 输出举例

```
→ GooglePixel5 fastboot boot magisk_patched-26100_tpJtT.img
Sending 'boot.img' (98304 KB) OKAY [ 2.447s]
Booting (bootloader) boot.img missing cmdline or OS version
OKAY [ 1.591s]
Finished. Total time: 4.077s
```

## 再用Magisk去永久写入 (patch后的boot.img)

- 概述: `Magisk -> 安装 -> 直接安装 (推荐) -> 开始 -> 重启`
- 详解
  - `Magisk -> 安装`
    -



- 直接安装 (推荐)

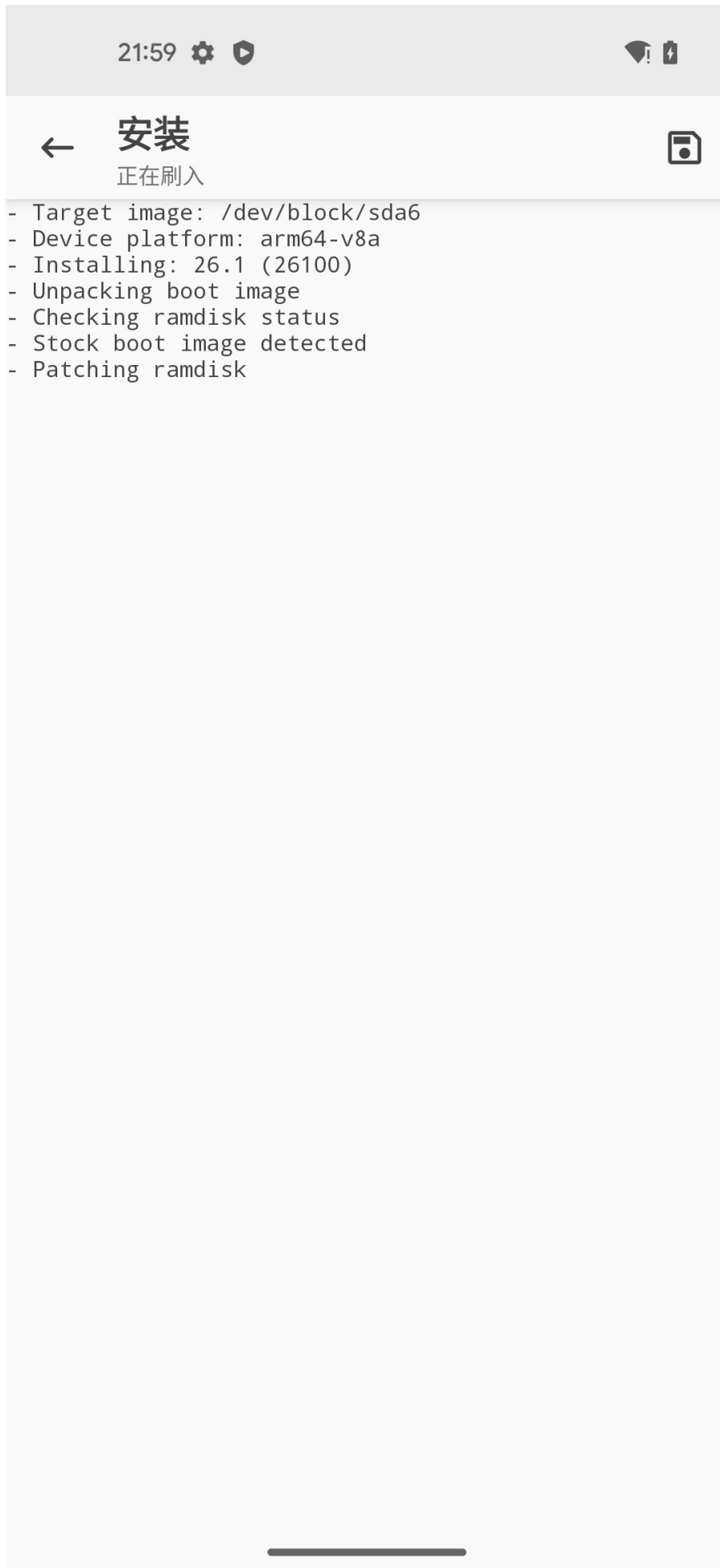






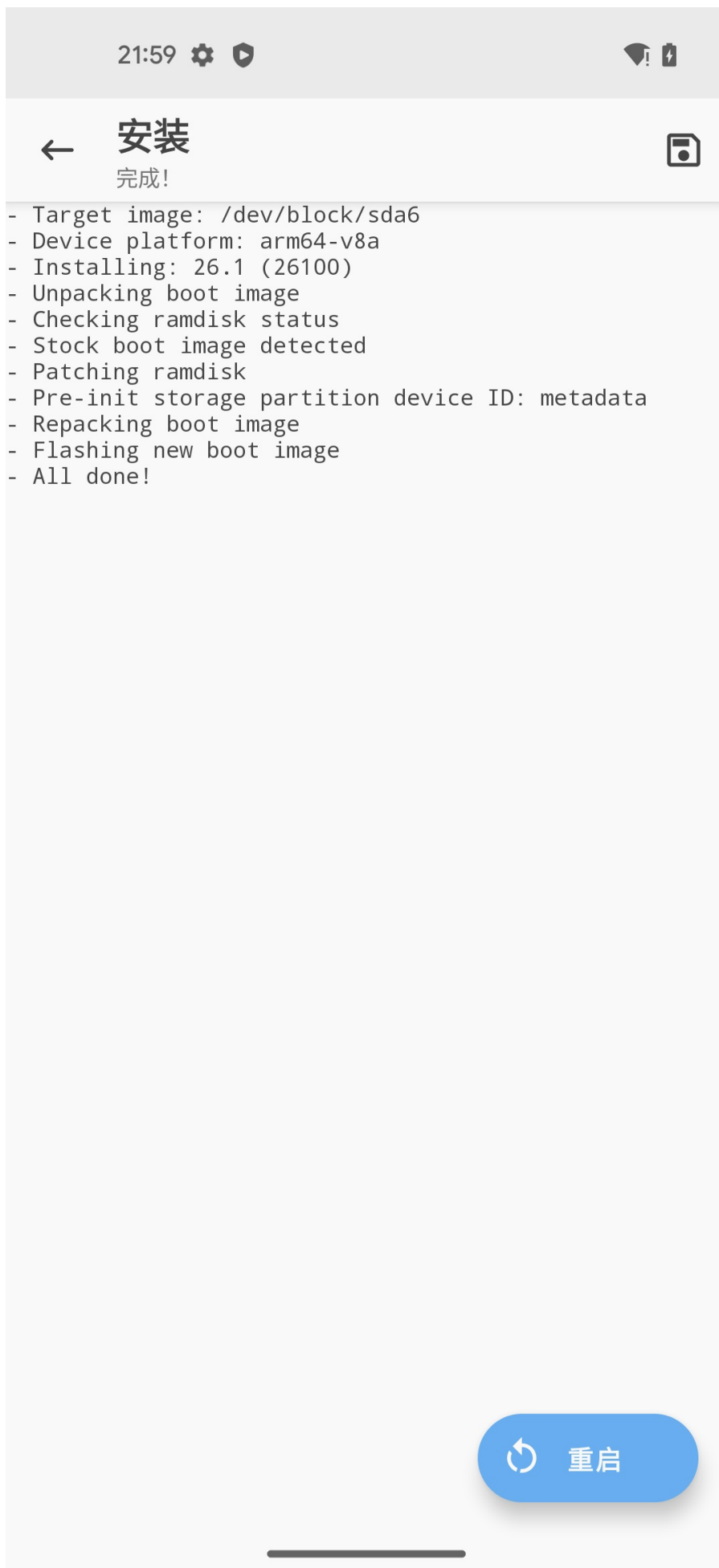
- 开始：正在刷入





- 重启



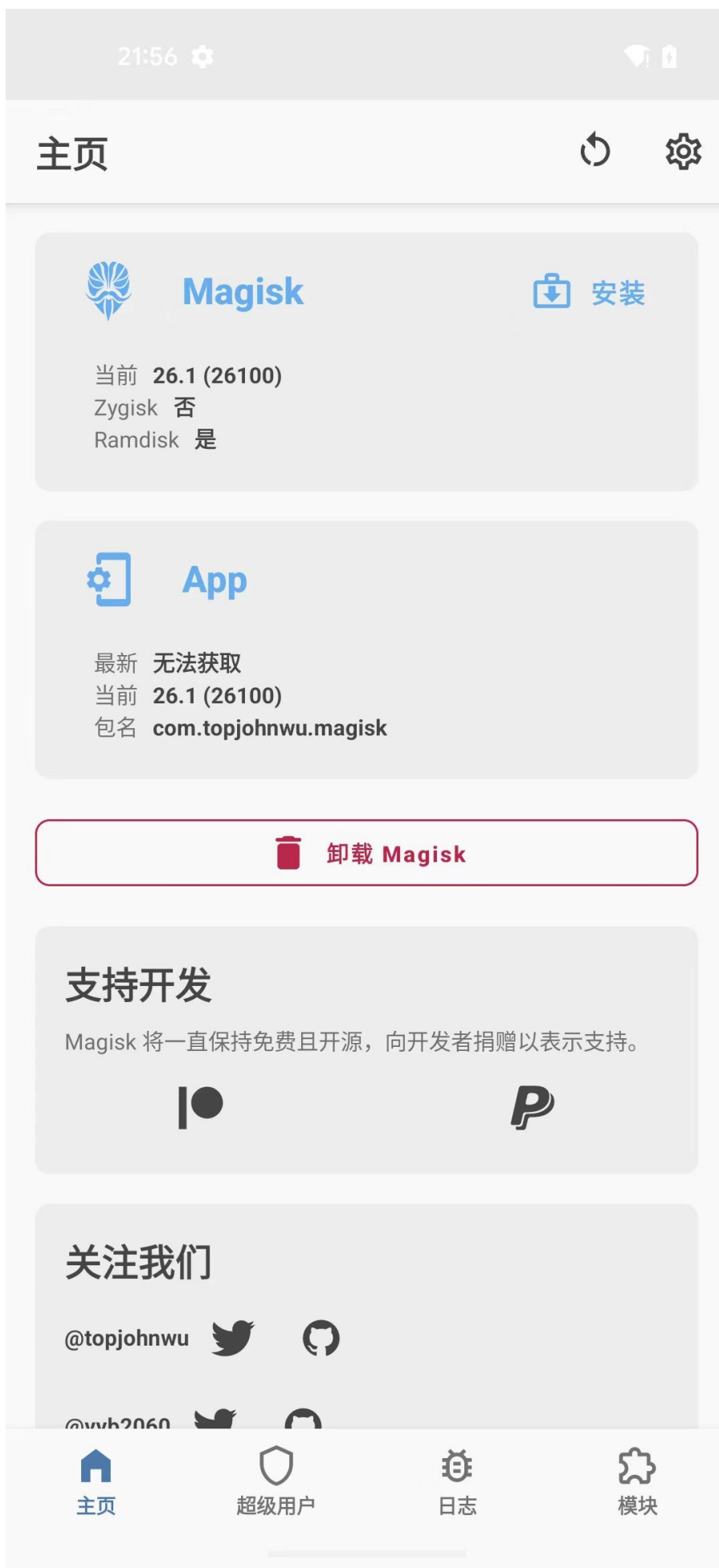


->效果:

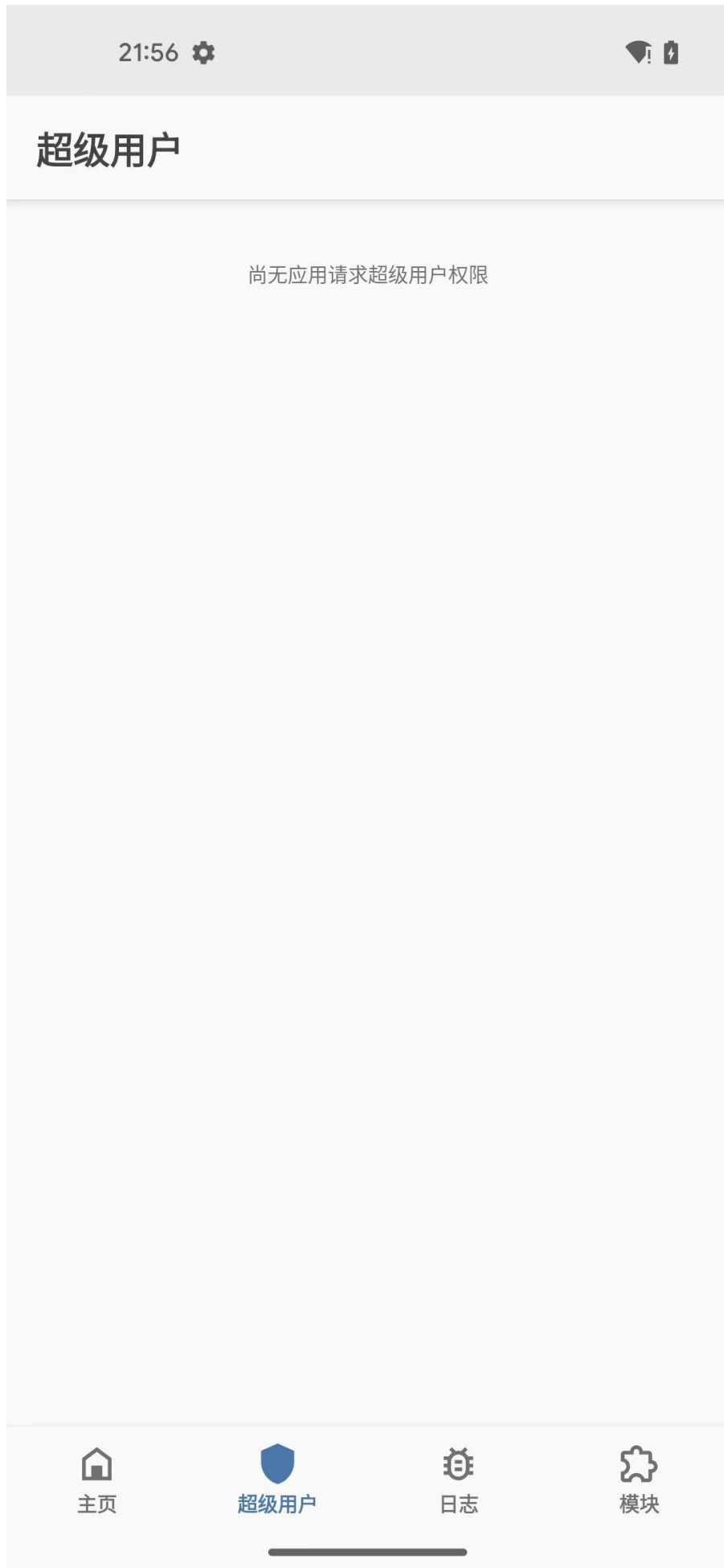
重启后, Magisk就拥有了root权限了

-> 就成功实现了, 用Magisk给Android13的Pixel5去root了

- 具体表现是: Magisk中的 超级用户 和 模块 2个tab页, 不是灰色。
  -



- 都可以点击进去了
  - 超级用户=root
    -





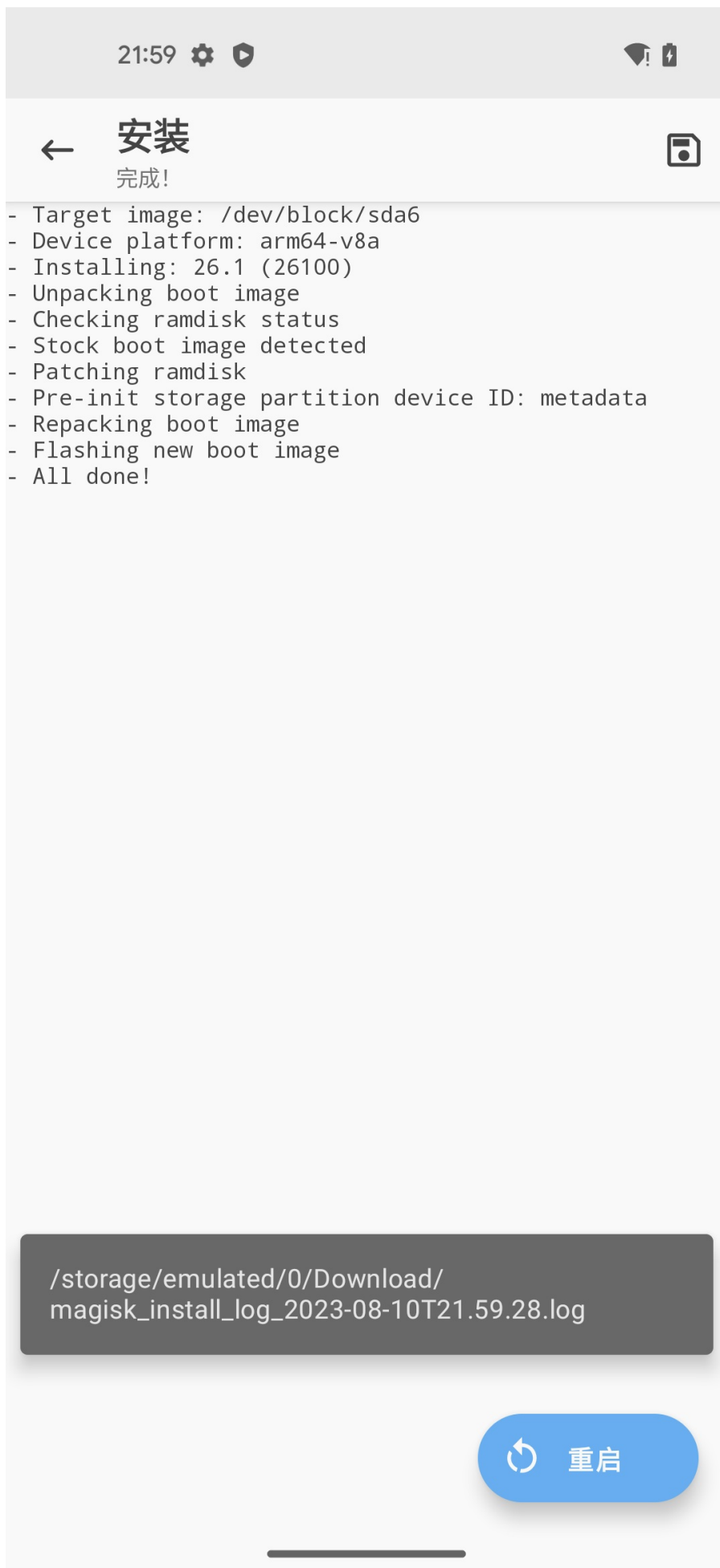
- 模块

-



## 附录：相关log日志

点击右上角的保存按钮：



可以保存出日志，导出到电脑：

```
→ GooglePixel5 adb pull /sdcard/Download/magisk_install_log_2023-08-10T21.59.28.log
/sdcard/Download/magisk_install_log_2023-08-10T21.59.28.log: 1 file pulled, 0 skipped. 0.3 MB/s (1942 bytes in 0.006s)
```

查看到日志内容：

- magisk\_install\_log\_2023-08-10T21.59.28.log

```
- Target image: /dev/block/sda6
- Device platform: arm64 v8a
- Installing 26.1 (26100)
Parsing boot image: [/dev/block/sda6]
HEADER_VER [3]
KERNEL_SZ [11721709]
RAMDISK_SZ [1987466]
OS_VERSION [13.0.0]
OS_PATCH_LEVEL [2023 08]
PAGESIZE [4096]
CMDLINE []
  Unpacking boot image
  KERNEL_FMT [lz4]
  RAMDISK_FMT [lz4_legacy]
  VBMETA
Loading cpio: [ramdisk cpio]
  Checking ramdisk status
  Stock boot image detected
  Patching ramdisk
  Pre init storage partition device ID metadata
Loading cpio: [ramdisk cpio]
Add entry [init] 0750
Create directory [overlay.d] 0750
Create directory [overlay.d/sbin] 0750
Add entry [overlay.d/sbin/magisk32.xz] 0644
Add entry [overlay.d/sbin/magisk64.xz] 0644
Add entry [overlay.d/sbin/stub.xz] 0644
Patch with flag KEEPVERITY [true] KEEPFORCEENCRYPT [true]
Loading cpio: [ramdisk.cpio.orig]
Backup mismatch entry: [init] -> [.backup/init]
Record new entry: [overlay.d] -> [.backup/.rmlist]
Record new entry: [overlay.d/sbin] -> [.backup/.rmlist]
Record new entry: [overlay.d/sbin/magisk32.xz] -> [.backup/.rmlist]
Record new entry: [overlay.d/sbin/magisk64.xz] -> [.backup/.rmlist]
Record new entry: [overlay.d/sbin/stub.xz] -> [.backup/.rmlist]
Create directory [.backup] 0000
Add entry [.backup/magisk] 0000
Dump cpio: [ramdisk.cpio]
Patch 014B4FC6 [738B69705F698E697472616D667300] -> [77616E745F698E697472616D667300]
Parsing boot image: [/dev/block/sda6]
HEADER_VER [3]
KERNEL_SZ [11721709]
RAMDISK_SZ [1987466]
OS_VERSION [13.0.0]
OS_PATCH_LEVEL [2023 08]
PAGESIZE [4096]
CMDLINE []
  Repacking boot image
  KERNEL_FMT [lz4]
  RAMDISK_FMT [lz4_legacy]
  VBMETA
Repack to boot image: [new boot.img]
HEADER_VER [3]
KERNEL_SZ [11586555]
RAMDISK_SZ [2513635]
OS_VERSION [13.0.0]
OS_PATCH_LEVEL [2023 08]
PAGESIZE [4096]
CMDLINE []
  Flashing new boot image
  All done
```

## 对比说明

注:

- 对比说明:
  - 之前:
    - **【未解决】** 给Android13的Pixel5刷入用Magisk去Patch后的boot.img
  - 的操作是:

```
fastboot flash boot magisk_patched-20100_bMrsR.img
```
  - 意思是:
    - 用Fastboot的flash一次性写入Magisk打了patch后的boot.img到boot分区
    - 以及后续还有个额外的动作: 绕过vbmeta的验证

```
fastboot flash vbmeta --disable-verity --disable-verification vbmeta.img
```
  - -》总之, 最后导致了:
  - 变砖卡死, 进入了Android Recovery Cannot load Android system的页面:
    - **【未解决】** 用Magisk给Pixel5去root重启报错: Android Recovery Cannot load Android system. Your data may be corrupt
  - 而无法恢复, 最终是:
    - **【已解决】** 安卓手机Pixel5变砖无法启动系统卡死在Fastboot Mode
    - **【已解决】** 尝试修复Pixel5卡死在Fastboot Mode: Android Flash Tool即flash.android.com
  - 去用:
    - [Android Flash Tool](#)
  - 最终重新刷回了官网的Android13的ROM, 而救砖成功的。

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: 2023-08-16 22:40:16

## Magisk中root相关设置

此时已经实现了用Magisk给安卓13的Pixel5去root，已获取到root权限了。

而后续其他内容想要去获取root权限，则就是通过Magisk去管理和授权了。

为了让其他内容更方便的获取root权限，则去改动Magisk中root相关的设置：

Magisk中，关于获取root权限的默认设置是：





- 超级用户
  - 自动响应: 提示
    - 别的内容, 想要请求root权限时, 是会弹框提示
      - 需要你自己手动去点击允许, 对方才能获取到root权限
  - 请求超时: 10秒
    - 如果不小心超时了(比如我之前就是为了截图记录过程, 而导致), 超过了此处的默认设置的10秒, 则表示拒绝了, 对方就无法获取到root权限了

去改为:



- 超级用户
  - 自动响应: 允许
    - 无需手动干预, 自动允许获取root权限
      - 则此时: 其实后续的参数: 请求超时 则就没意义=不起作用了=用不到这个设置了
  - 请求超时: 60秒
    - 把超时时间改的足够的长, 方便能来得及操作

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: 2023-08-16 22:47:24

## 给LineageOS19的的小米8去root

此处有个：

刷机=ROM改机，为：`LineageOS 19` 的 `Android 12` 的 小米8：

- 小米8
  - 核心参数
    - LineageOS 19
    - Android 12
  - 图
    -



■




去用（新版Magisk的）小狐狸 = 狐狸面具 = Kitsune Mask 去root。

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: 2024-11-14 09:33:26



# 下载安装Kitsune Mask

## 下载Kitsune Mask

- (新版Magisk的) Kitsune Mask
  - 别称
    - 小狐狸
    - 狐狸面具
  - logo
    - 
  - 源码
    - <https://github.com/KitsuneMagisk/magisk>
  - 下载
    - 稳定版
      - <https://github.com/HuskyDG/download/>
        - v26.4 Stable
          - <https://github.com/HuskyDG/download/raw/main/magisk/26.4-kitsune.apk>
    - 最新版
      - <https://github.com/HuskyDG/magisk-files>
        - <https://github.com/HuskyDG/magisk-files/releases/>
          - 27001
            - <https://github.com/HuskyDG/magisk-files/releases/tag/1720171347>
              - debug版
                - <https://github.com/HuskyDG/magisk-files/releases/download/1720171347/app-debug.apk>
              - release版
                - <https://github.com/HuskyDG/magisk-files/releases/download/1720171347/app-release.apk>

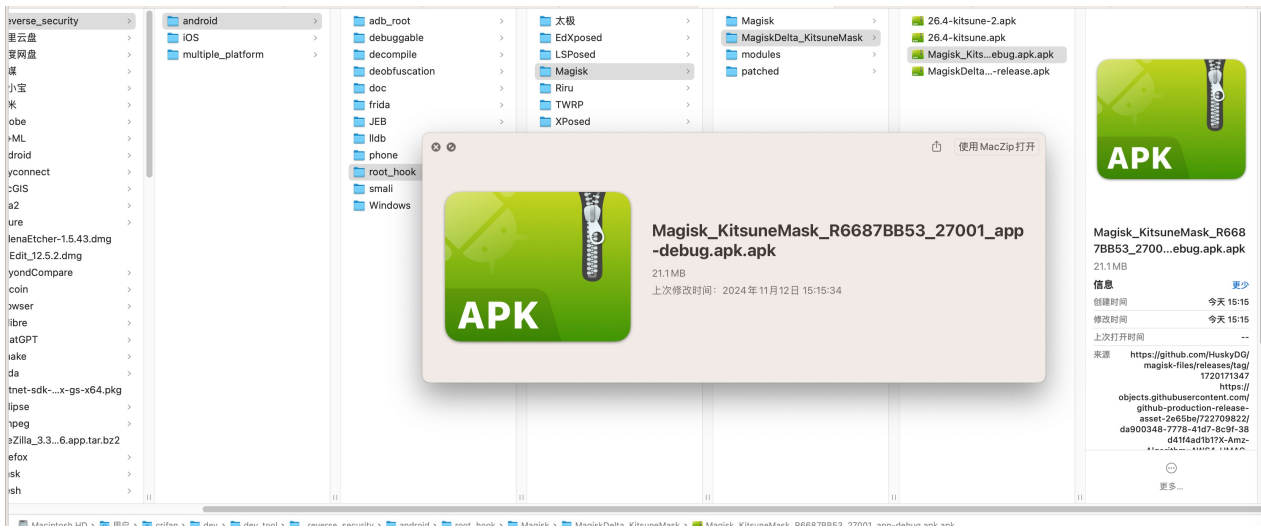
下载任何一个版本均可

比如，此处下载了：

<https://github.com/HuskyDG/magisk-files/releases/download/1720171347/app-release.apk>

且改名为：

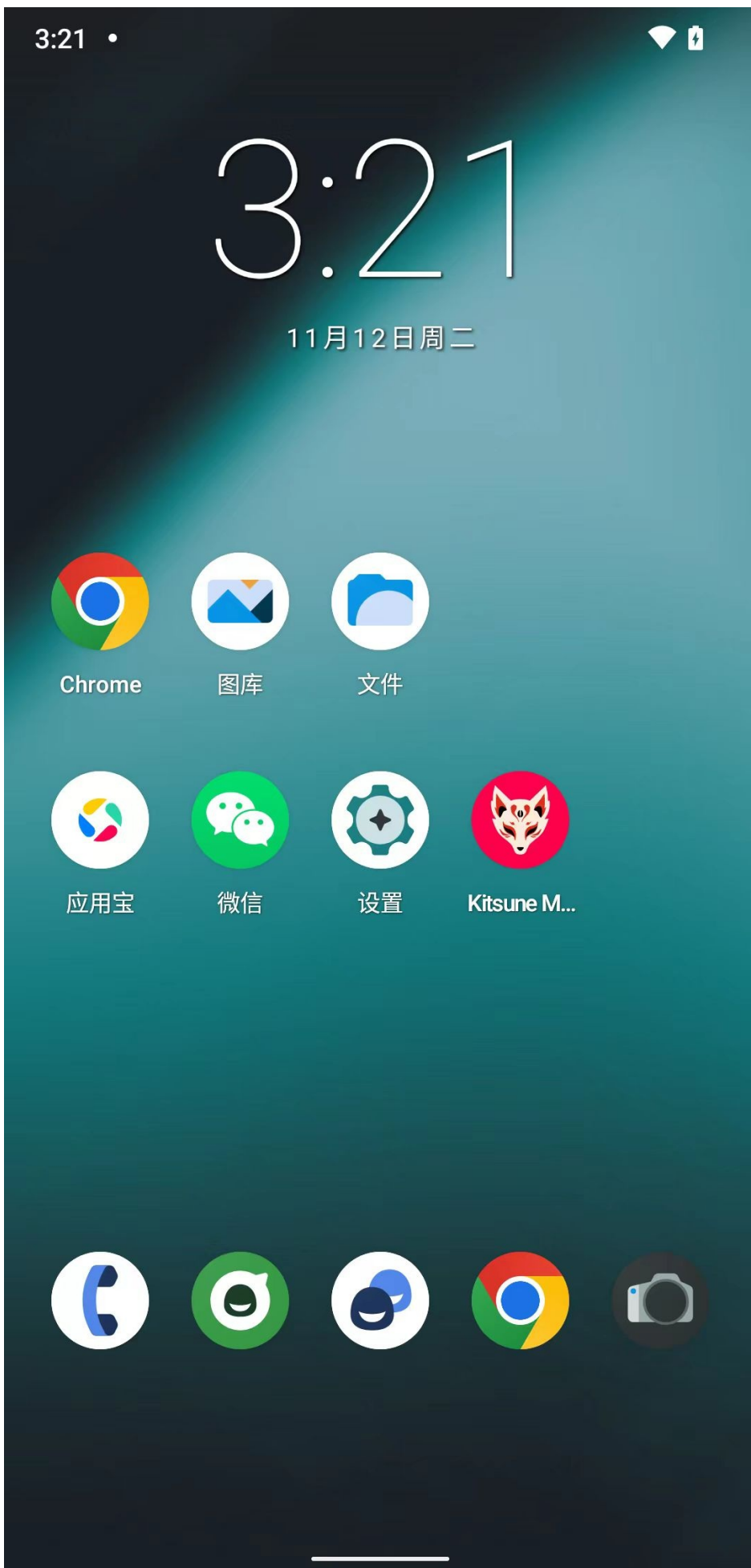
Magisk\_KitsuneMask\_R6687BB53\_27001\_app-debug.apk.apk



## 安装Kitsune Mask

```
adb install Magisk_KitsuneMask_R6687BB53_27001_app-debug.apk
```

安装后，桌面中出现 Kitsune Mask 的图标：



## 打开Kitsune Mask

安装后，去打开：

3:21 •



# 主页



警告！您正在使用 Magisk 的非官方版本，该版本未得到 Magisk 作者的正式支持。此版本可能包含不稳定的修改，可能会损害您的设备或危及您的安全。请勿报告任何问题或向 Magisk 官方渠道请求任何帮助。如果您使用此版本但不知道这是非官方版本，请切换到官方 Magisk [github.com/topjohnwu/Magisk](https://github.com/topjohnwu/Magisk)

不再显示



## Magisk

安装

当前 无法获取  
 Zygisk 否  
 Ramdisk 是



## App

安装

最新 R6687BB53-kitsune (27001) (D)  
 当前 R6687BB53-kitsune (27001) (D)  
 包名 io.github.huskydg.magisk

## 支持开发

Magisk 将一直保持免费且开源，向开发者捐赠以表示支持。



## 关注我们



主页



超级用户



日志



模块

点击：不再显示，后：



其中：

- 超级用户 和 模块：是灰色的 == 不可点击的 == 表示含义：还没开启root权限

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：2024-11-14 10:46:54



## 解锁Bootloader

此处，按理说，应该也去参考：

普通的安卓手机去root之前的：

[解锁Bootloader](#)

不过，此处有点特殊：

- 原先二手已root小米8
  - 开发者选项中有：OEM解锁 -》且已解锁
    - 注：网上买二手时，卖家已帮忙实现 BL解锁
    - 图
    -

10:07 

   92



# 开发者选项

开启开发者选项



桌面备份密码

桌面完整备份当前未设置密码保护



不锁定屏幕

充电时屏幕不会休眠



直接进入系统

开启后不使用系统的锁屏样式，点亮屏幕直接进入系统，需要无屏幕密码



打开蓝牙数据包日志

已停用

已停用 

OEM 解锁

引导加载程序已解锁



WebView 实现

Android System WebView



设备解锁状态

查看当前设备的解锁状态

已解锁 

- 但是后来把 MIUI 的小米8刷机为 LineageOS 后
  - 开发者选项中（竟然）：没有OEM解锁
    - 感觉是：无需OEM解锁，所以算：已解锁
    - 图
    -



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: 2024-11-14 10:50:03

## 找到要打补丁的boot.img文件

### 要打补丁的文件是哪个文件

- 要打补丁的文件是哪个文件
  - 根据官网[Installation | Magisk](#)的解释，其实底层细节逻辑是：
  - 此处
    - Kitsune Mask中Magisk中：`ramdisk =` 是
    -



- 表示：支持 `Ramdisk`
- 所以，要打补丁的文件，是： `boot.img`

## 到哪里找boot.img

去你的系统的镜像=image中，找对应的 `boot.img`

而此处要打补丁的 `boot.img` 文件，是

之前（把 `MIUI` 的 小米8 ）刷机为 `LineageOS 19` 时所用的ROM包：

- `lineage-19.1-20241110-UNOFFICIAL-dipper.zip`

◦

去解压可以得到其中的：

- （ `67.1MB` 的） `boot.img`

◦

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：2024-11-14 10:56:31



## 用Kitsune Mask去给boot.img打patch

此处：用Kitsune Mask去给boot.img打patch

和之前的：

[用Magisk去给boot.img打patch](#)

的操作，其实可以说是：一模一样的

此处，仍记录细节过程，供参考：

---

把 boot.img 下载到安卓手机小米8中：

```
→ lineage-19.1-20241110-UNOFFICIAL-dipper adb push boot.img /sdcard/Download/  
boot.img: 1 file pushed, 0 skipped. 106.8 MB/s (67092480 bytes in 0.599s)
```

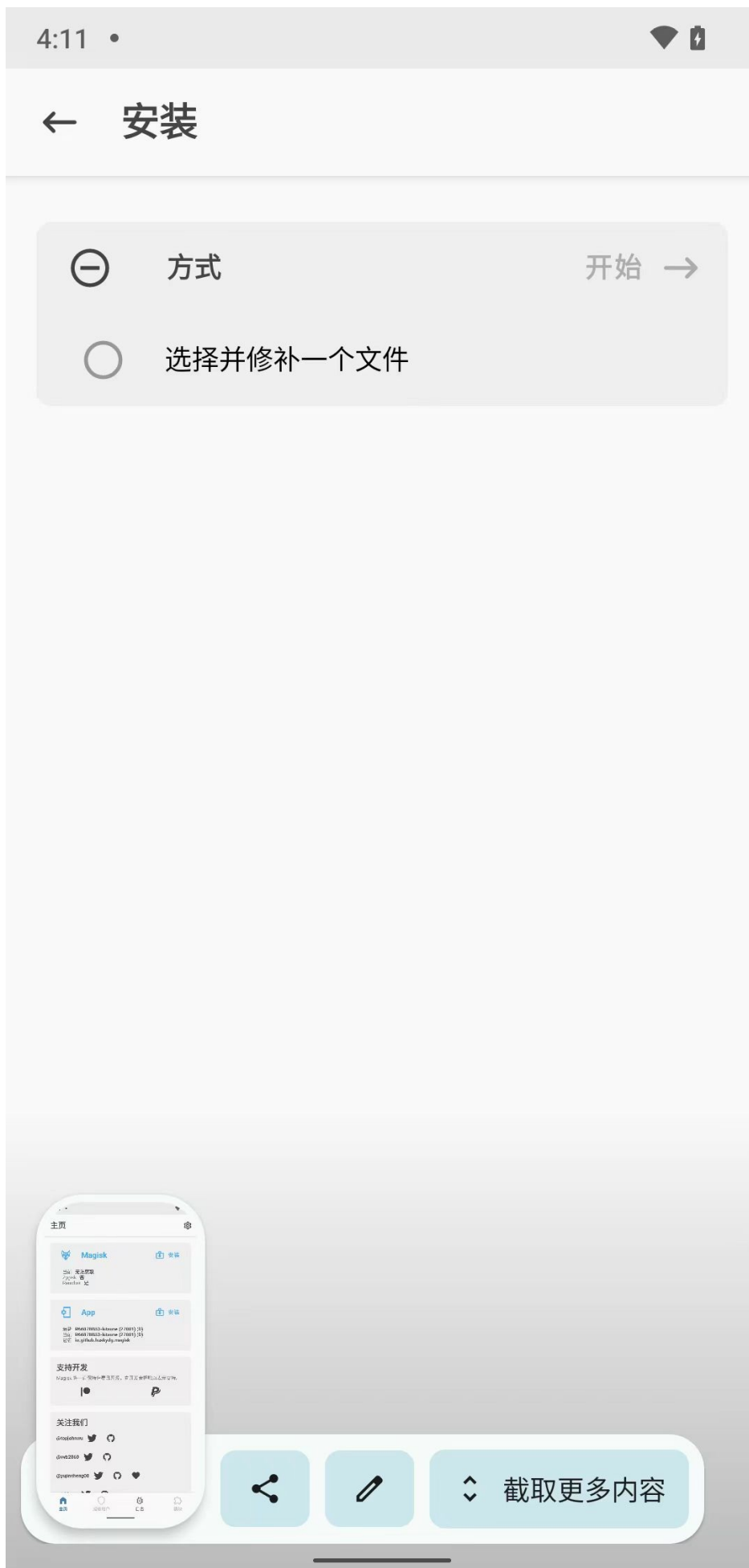
去小米8中打开 Kitsune Mask ：

点击 Magisk 的： **安装** （注意：不是 App 的 安装）：

-



点击：[选择并修补一个文件](#)：



弹框选择（之前已从电脑端下载到手机端下载目录中的）文件：`boot.img`：

4:12 •



≡ 下载



下载



“下载”中的文件

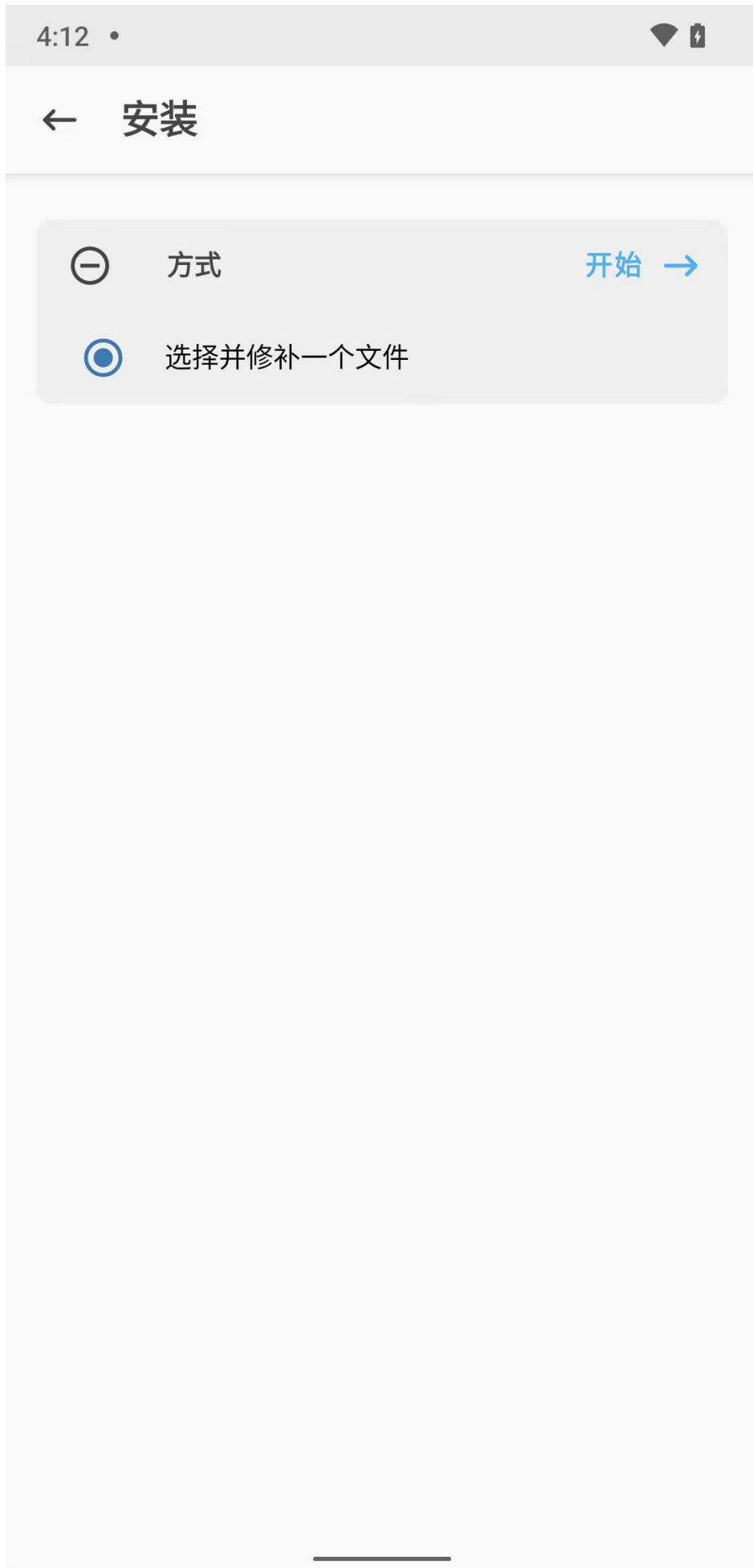


boot.img

2009年1月1日, 67.09 MB, BIN 文件

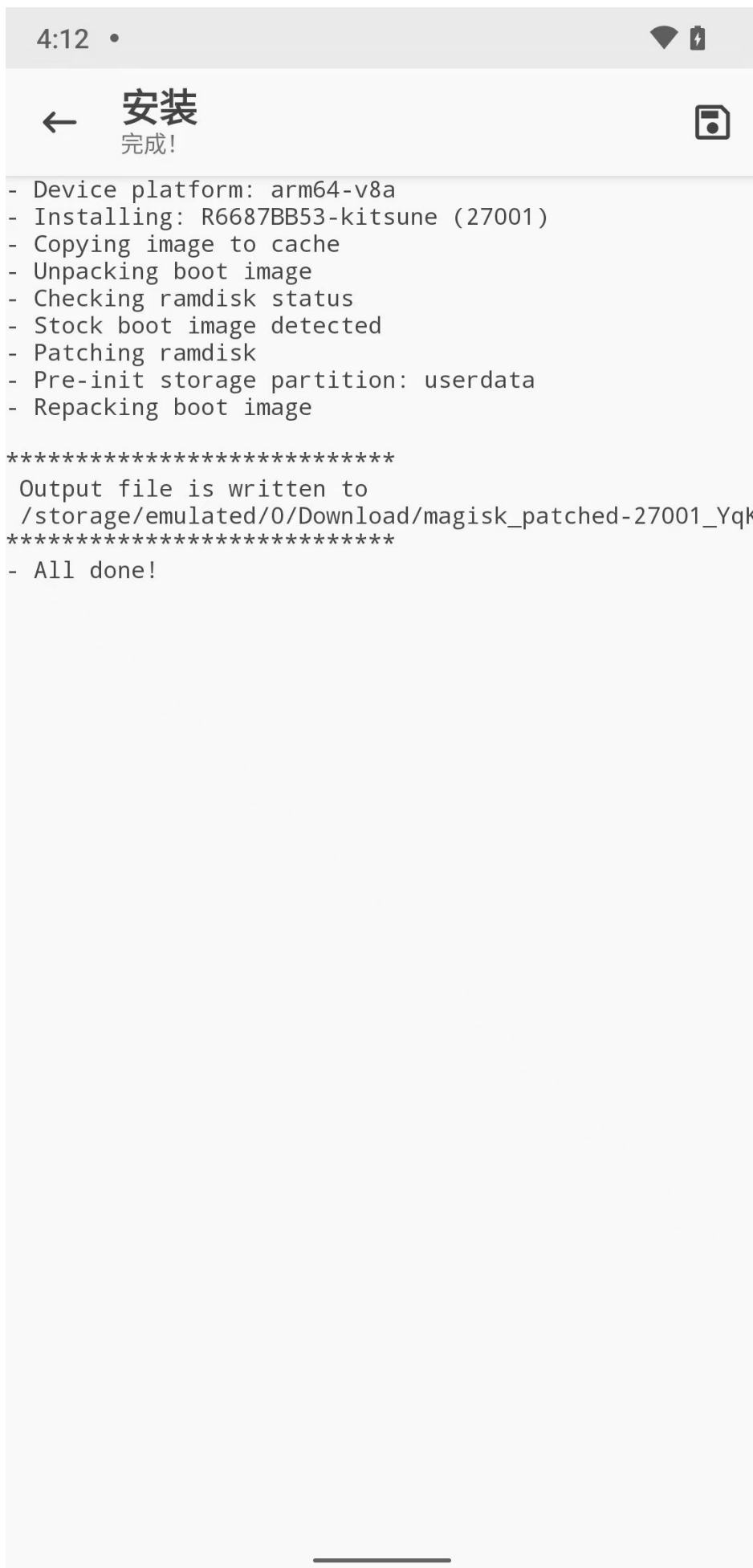


已选择了 [选择并修补](#) 一个文件， 点击 [开始](#) ->





开始安装过程，显示进度日志。最后显示 `All done` ，表示完成：







可以看到输出的文件 = 打了补丁的 boot.img 文件是：

```
/storage/emulated/0/Download/magisk_patched-27001_YqKbC.img
```

去看看文件细节：

```
dipper:/ $ cd /sdcard/Download/
dipper:/sdcard/Download $ ll
/system/bin/sh: ll: inaccessible or not found
127 dipper:/sdcard/Download $ ls -lh
total 128M
-rw-rw---- 1 root everybody 64M 2009-01-01 00:00 boot.img
-rw-rw---- 1 root everybody 64M 2024-11-12 16:12 magisk_patched-27001_YqKbC.img
```

继续上传到PC电脑端：

```
→ magisk_kitsuneMask_root adb pull /sdcard/Download/magisk_patched-27001_YqKbC.img .
/sdcard/Download/magisk_patched-27001_YqKbC.img: 1 file pulled, 0 skipped. 35.7 MB/s (67092480 bytes in 1.795s)
→ magisk_kitsuneMask_root ll
total 131456
-rw-r--r-- 1 crifan staff 64M 11 12 16:25 magisk_patched-27001_YqKbC.img
```

供下一步 [用Kitsune Mask](#)写入patch后的boot.img 去使用。

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: 2024-11-14 11:01:08

## 用Kitsune Mask写入patch后的boot.img

此处的：用Kitsune Mask写入patch后的boot.img

其实也是和之前的：

[用Magisk写入patch后的boot.img](#)

的操作：一模一样。

但是也是额外记录此处细节，供参考：

---

## 进入Bootloader模式

电脑端：

```
adb reboot bootloader
```

此时小米8手机中进入 FASTBOOT 界面：



再去用 `fastboot devices` 确认:

```
→ magisk_kitsuneMask_root fastboot devices
f97a0408 fastboot
```

## 写入打了补丁后的boot.img文件

然后继续去用 `fastboot boot magisk_patched-27001_Yqkbc.img` 写入 (之前打了补丁的) `boot.img` 文件:

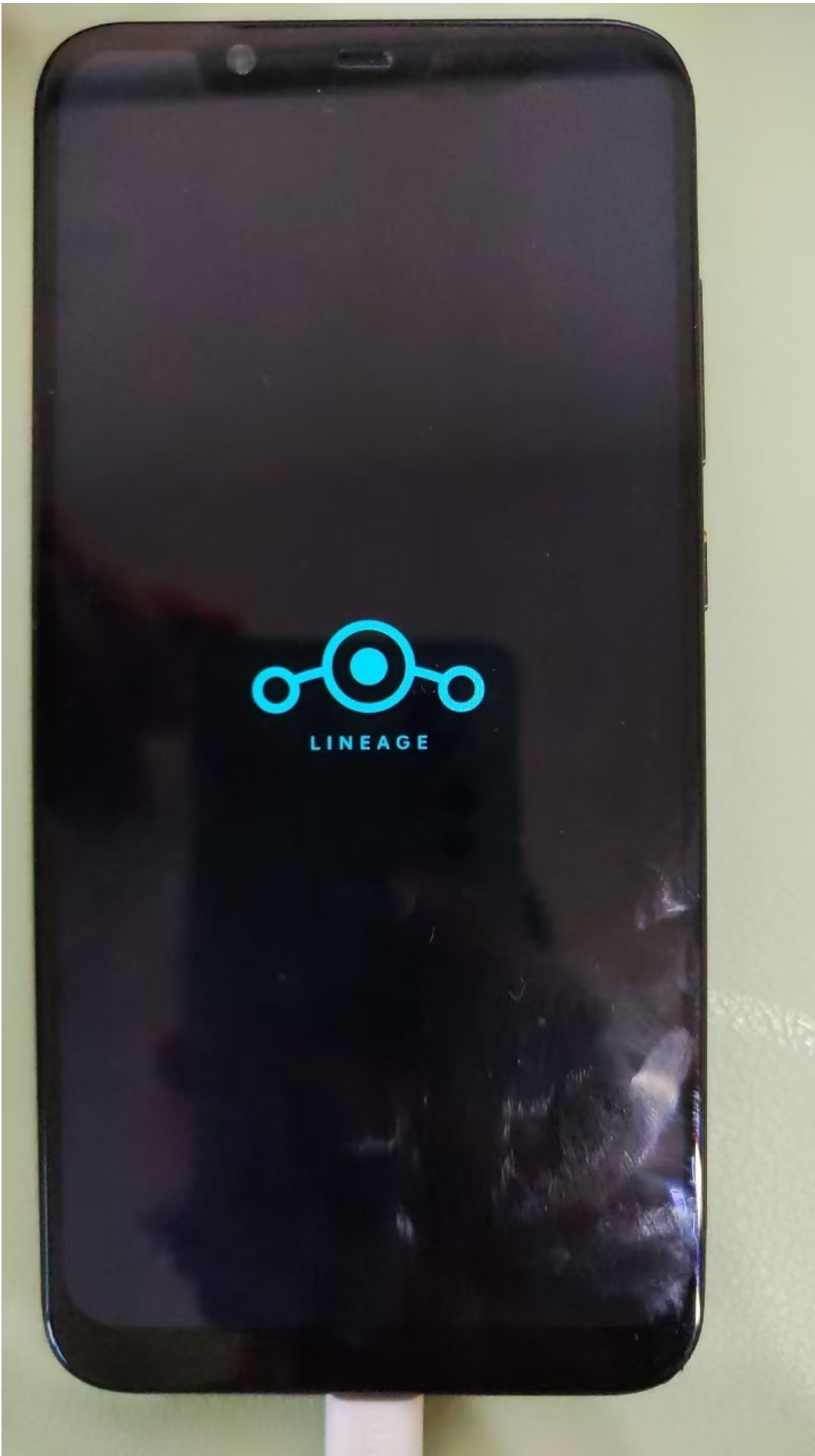
效果:

```
→ magisk_kitsuneMask_root fastboot boot magisk_patched-27001_Yqkbc.img
Sending 'boot.img' (65520 KB)          OKAY [ 1.516s]
Booting                               OKAY [ 0.050s]
Finished. Total time: 1.580s
```

## 手机自动重启

然后手机端小米8会重启:





重启后，进入桌面：

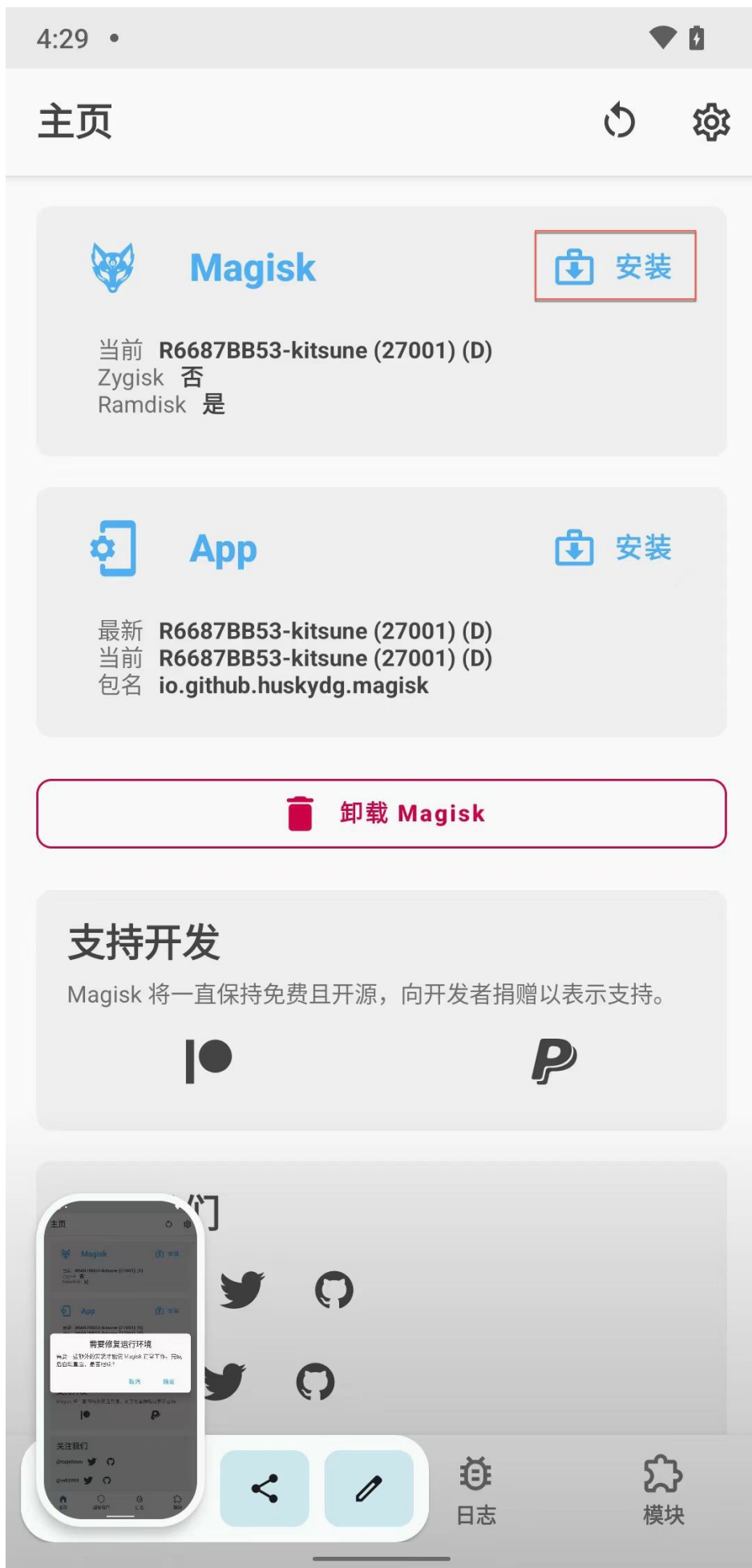


## 进入Kitsune Mask

打开 Kitsune Mask :

- 此时往往要注意和处理:
  - [弹框: 需要修复运行环境](#)

继续点击 Magisk 后面的 安装 :



## 直接安装

点击：[直接安装 \(推荐\)](#)



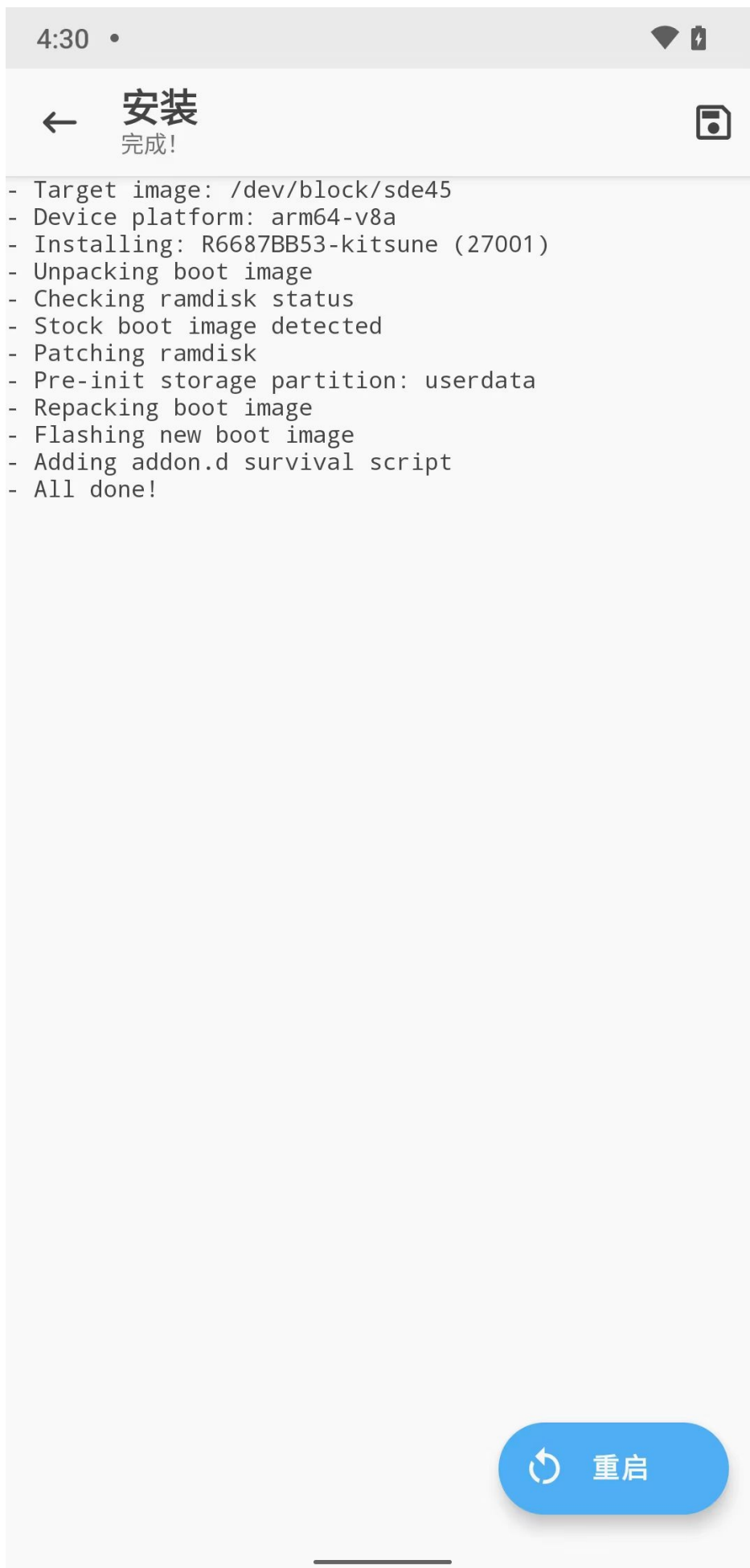
已勾选  直接安装 (推荐) ， 点击：





## 再次重启手机

开始安装过程。稍等几十秒后，完成。点击按钮：[重启](#)





## 弹框：需要修复运行环境

- 概述
  - 对于 需要修复运行环境 ， 点击 取消
    - 注意：不要点击 确定

下面是详细解释：

---

对于刚用：

```
fastboot boot magisk_patched-27001_YqKbC.img
```

写入打了补丁的 boot.img 后（手机重启后），再去打开 Kitsune Mask 时，往往会遇到一个稍微特殊点的情况：

会出现弹框提示： 需要修复运行环境

```
需要修复运行环境
需要一些额外的安装才能使Magisk正常工作。完成后自动启动，是否继续
取消
确定
```



## 【总结】

根据实测的经验：

- Kitsune Mask
  - 不论是之前的，最新版，测试版，不是稳定版： `R65C33E4F-kitsune (27001)`
  - 还是官网推荐的，最新的稳定版： `26.4-kitsune (26400)`

对于，如果用fastboot去启动patch后的boot.img：

```
fastboot boot magisk_patched-27001_02xja.img
```

(重启后) 然后：

启动Kitsune Mask： 都会出现：

- 需要修复运行环境
  - 需要一些额外的安装才能使Magisk正常工作。完成后自动重启，是否继续？
    - 取消 确定

这个弹框，而：

- 如果点击： `确定` => 就会导致（期望的，已经获取到的，临时的）**root**权限丢失掉
  - 超级用户 和 模块： 2个tab按钮，灰色，不可点击
    - 后续也就无法正常root流程了
- 应该是点击： `取消` => 忽略这个提示 => 即可实现：（临时的）**root**权限得以保留
  - 此时： `超级用户` 和 `模块`： 2个tab按钮，是白色的，可点击的
  - 然后继续后续步骤：
    - [用Kitsune Mask写入patch后的boot.img](#)
  - 即可顺利完成root

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： 2024-11-14 11:05:02

## Kitsune Mask中root相关设置

同理，`Kitsune Mask` 和 `Magisk` 中的配置和逻辑：

[Magisk中root相关设置](#)

几乎一样。

但也还是再仔细介绍一下：



---

正常进入已root后的 `Kitsune Mask` 的界面是：





4:34 • 📶 🔋


主页 🔄 ⚙️

 **Magisk**  安装

当前 **R6687BB53-kitsune (27001) (D)**  
Zygisk 否  
Ramdisk 是



 **App**  安装

最新 **R6687BB53-kitsune (27001) (D)**  
当前 **R6687BB53-kitsune (27001) (D)**  
包名 **io.github.huskydg.magisk**



 **卸载 Magisk**



**支持开发**


Magisk 将一直保持免费且开源，向开发者捐赠以表示支持。


 


**关注我们**


@topjohnwu  

@vvb2060  

  
主页

  
超级用户

  
日志

  
模块

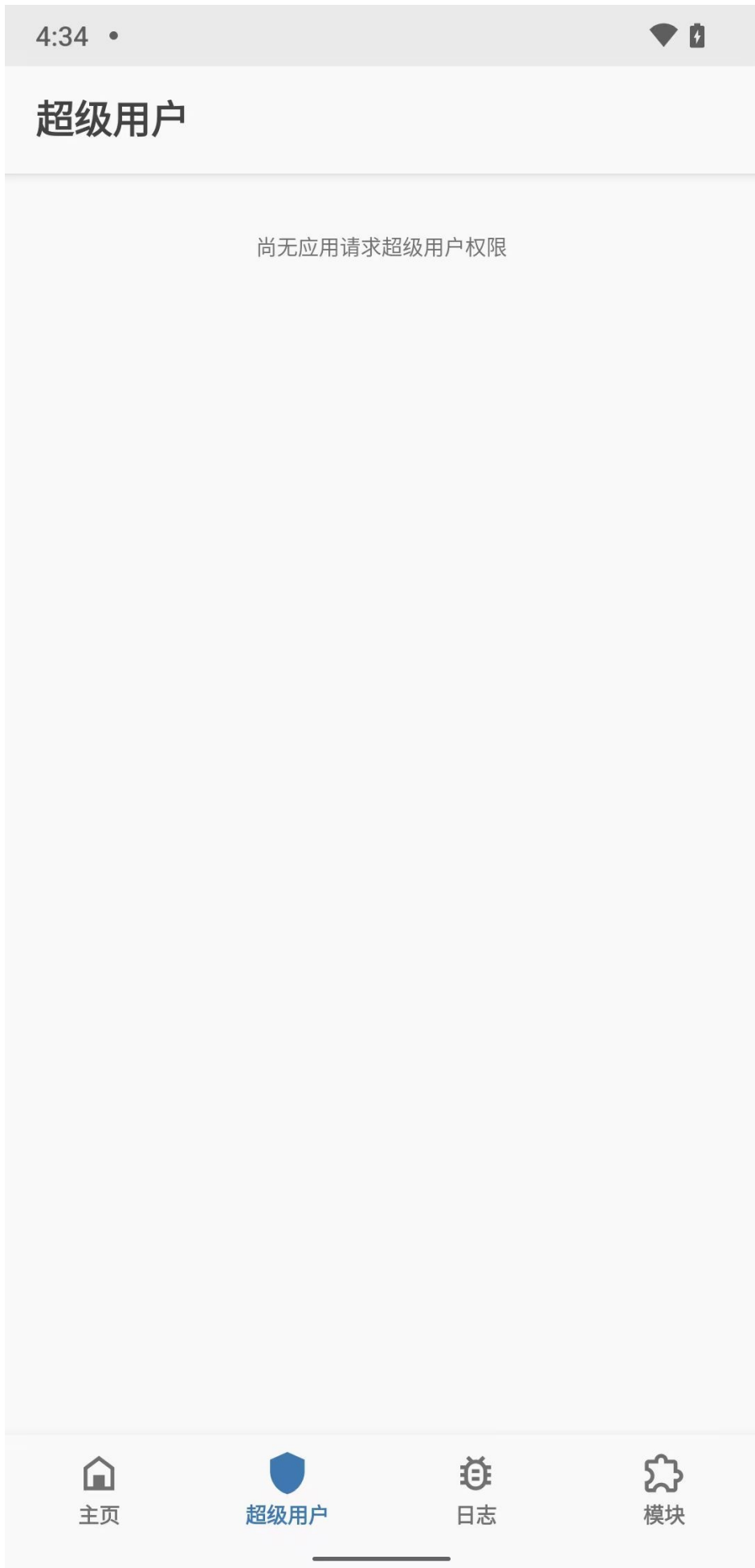
常规的逻辑和功能:

## 如何判断是否已经root成功?

超级用户 和 模块 : 不是灰色, 是白色, 且可以点击, 表示: 已获取了root权限

## 超级用户

- 超级用户
  -



## 模块

- 模块
  -



## 设置

- 设置
  - 进入方式：点击 Kitsune Mask 主页的右上角**齿轮**按钮
  - 默认的一些设置
    -



-



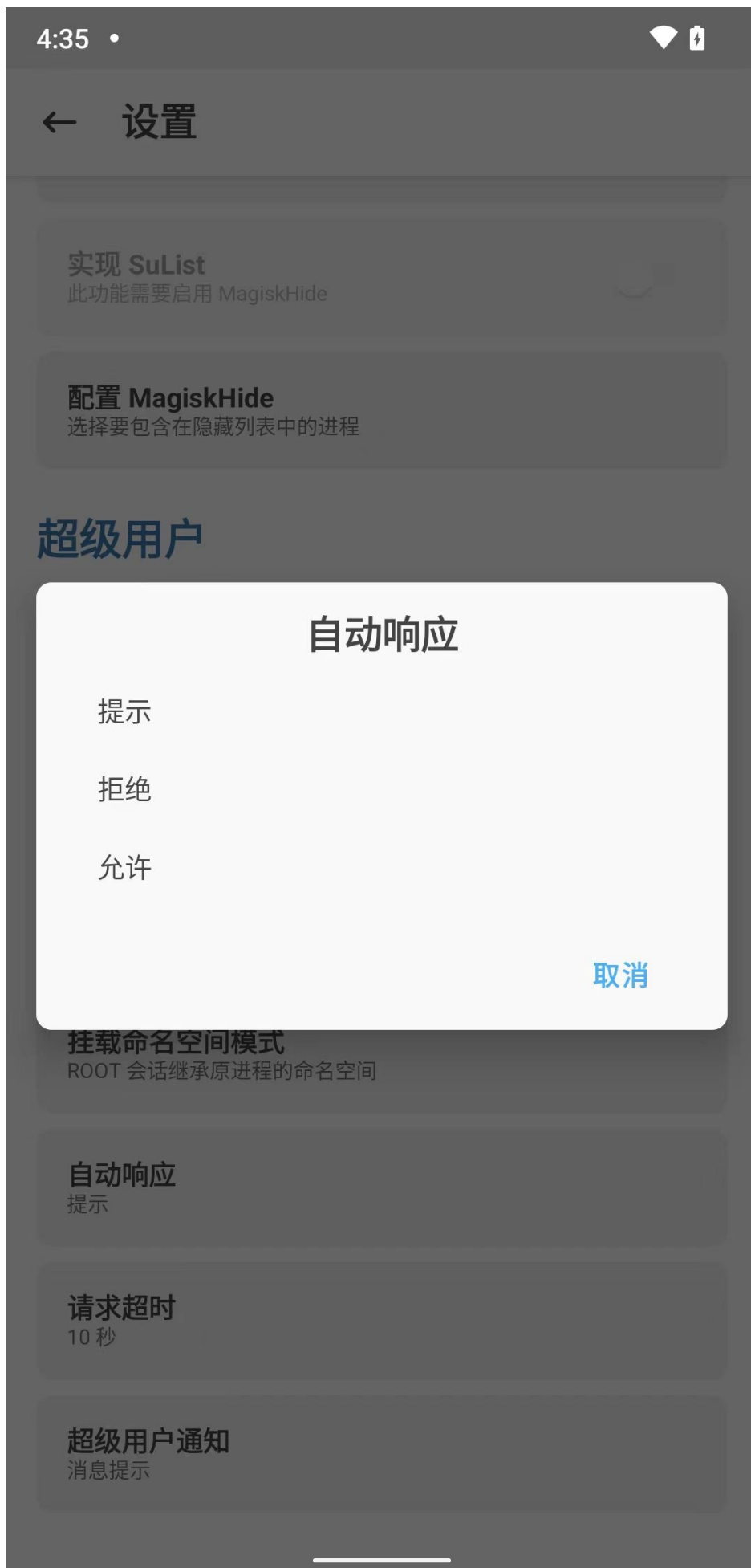


-



## 改动一些配置

- 自动响应
  - 之前: 提示
    - 图
      -



- 效果和逻辑：别人应用申请获取root后，此处会弹框提示，需要用户手动点击允许，才能真正获取root权限 =》比较麻烦
- 现在改为： (自动) 允许
  - 图
    -



- 效果和逻辑：别人应用申请获取root后，自动就同意，无需用户手动同意，更加方便

## 日志

- 日志
  -



4:34 •
📶 🔋

## 日志

📄 🗑️

```

01-11 04:24:47.836 809 809 I : Magisk R6687BB53-ki
01-11 04:24:47.836 809 809 D : resetprop: prop [ro
01-11 04:24:47.838 809 809 I : * Device API level:
01-11 04:24:47.838 809 809 D : resetprop: get prop
01-11 04:24:47.838 809 809 D : resetprop: get prop
01-11 04:24:47.838 809 809 I : * CPU ABI 64-bit: a
01-11 04:24:47.838 809 809 I : * CPU ABI 32-bit: a
01-11 04:24:47.847 809 811 D : resetprop: get prop
01-11 04:24:47.847 809 811 D : resetprop: get prop
01-11 04:24:47.854 809 811 I : ** post-fs-data mod
01-11 04:24:47.854 809 811 I : * Magic Mount Setup
01-11 04:24:48.174 809 811 I : * Initializing Magi
01-11 04:24:48.229 809 811 D : resetprop: decode w
01-11 04:24:48.238 809 811 D : resetprop: prop [pe
01-11 04:24:48.238 809 811 D : resetprop: prop [ro
01-11 04:24:48.238 809 811 I : * Running post-fs-d
01-11 04:24:48.258 809 811 I : * Running module po
01-11 04:24:48.258 809 811 I : * Loading modules
11-12 16:33:33.002 809 811 I : ** late_start servi
11-12 16:33:33.002 809 811 I : * Running service.d
11-12 16:33:33.013 809 811 I : * Running module se
11-12 16:34:00.323 809 811 I : ** boot-complete tr
11-12 16:34:00.323 809 811 I : * Mount MagiskSU
11-12 16:34:00.328 809 811 D : tmpfs : /debug_ra
11-12 16:34:00.329 809 811 D : cp_link : /debug_ra
11-12 16:34:00.329 809 811 D : mirror : /debug_ra
11-12 16:34:00.329 809 811 D : mirror : /debug_ra
11-12 16:34:00.329 809 811 D : mirror : /debug_ra
11-12 16:34:00.329 809 811 D : cp_link : /debug_ra
11-12 16:34:00.330 809 811 D : mirror : /debug_ra
11-12 16:34:00.330 809 811 D : mirror : /debug_ra
11-12 16:34:00.330 809 811 D : cp_link : /debug_ra
11-12 16:34:00.330 809 811 D : mirror : /debug_ra
11-12 16:34:00.330 809 811 D : mirror : /debug_ra
11-12 16:34:00.330 809 811 D : mirror : /debug_ra
11-12 16:34:00.330 809 811 D : mirror : /debug_ra
11-12 16:34:00.330 809 811 D : mirror : /debug_ra
11-12 16:34:00.330 809 811 D : mirror : /debug_ra
11-12 16:34:00.330 809 811 D : mirror : /debug_ra
11-12 16:34:00.330 809 811 D : mirror : /debug_ra
11-12 16:34:00.331 809 811 D : mirror : /debug_ra
11-12 16:34:00.331 809 811 D : mirror : /debug_ra
11-12 16:34:00.331 809 811 D : mirror : /debug_ra
11-12 16:34:00.331 809 811 D : cp_link : /debug_ra
11-12 16:34:00.331 809 811 D : cp_link : /debug_ra
11-12 16:34:00.331 809 811 D : mirror : /debug_ra
11-12 16:34:00.331 809 811 D : mirror : /debug_ra

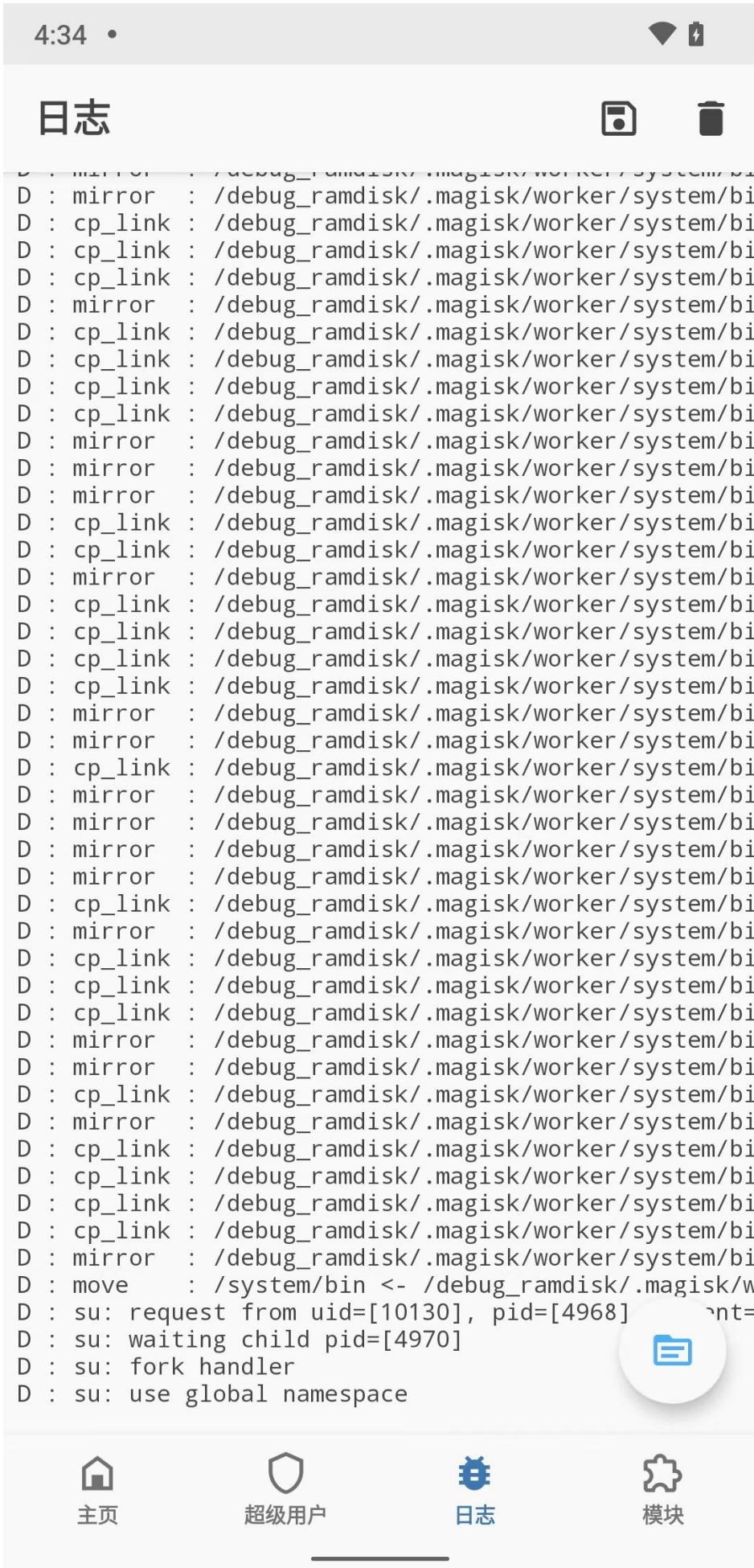
```

🏠 主页
🛡️ 超级用户
🐛 日志
⚙️ 模块

-



-



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: 2024-11-14 11:07:22

## root相关

TODO:

- 【未解决】 安卓逆向：尝试用adb-insecure给adb的shell增加root权限
- 【已解决】 安卓设备Google Pixel3中获取root权限使得su不报错Permission denied
- 【记录】 adb没有root权限就无法正常工作的相关现象
- 【已解决】 adb root报错：adb cannot run as root in production builds
- 【未解决】 adb shell中root和su都没有权限修改Download目录下的apk文件的权限属性
- 【未解决】 root的安卓手机Google Pixel3不稳定

---

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: 2022-10-30 17:18:30

## A/B槽位

TODO:

- **【已解决】** Magisk版本升级选项: 安装到未使用的槽位 (OTA后)

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: 2022-10-30 17:16:24



## OPPO R11s

TODO:

- 【未解决】用EDL模式和高通芯片烧录工具去烧录OPPO R11s的ROM
- 【已解决】安卓手机EDL模式
- 【未解决】用Windows电脑试试奇兔刷机能否给OPPO R11s解锁或刷入TWRP的Recovery
- 【记录】OPPO R11s重启后进入了ColorOS自带Recovery模式
- 【未解决】寻找可用的免授权的OPPO R11s的下载烧录工具

---

## OPPO R11s的ROM

TODO:

- 【已解决】从OPPO R11s的ozip导出的zip中提取boot.img文件

## OPPO R11s的root

像 `OPPO R11s`，如果不小心，重新安装了官方的rom，则就丢失了 `Bootloader` 的解锁

-> 就没法重新刷TWRP等第三方工具了，就没法正常继续root了

-> 只能想办法再去重新解锁：

要么寄回卖家重新帮你解锁

要自己弄：就涉及到手机中的芯片，比如高通的，相关刷机工具。

主要是烧录数据到Flash中的相关一套工具

而这些工具往往是需要签名授权验证的才能用的

一般不太容易找到免费的下载，而可能要找别人网上解锁，是要收费的

自己之前找了相关工具，但是后来还没精力继续尝试，所以暂时不确定：网上是否能找到，完全的免费的高通的刷固件的工具。

另外，期间涉到的OPPO的官网ROM，倒是可以找到免费的。

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：2022-10-30 17:54:06

## root心得

关于安卓的root心得:

- 安卓root
  - 安卓模拟器: 一般都很简单
    - 一般的模拟器都支持直接开启root
  - 安卓手机
    - 很早之前: 很容易
      - 尤其是 Android < 4.0的时代
        - 随便去买个手机, 都能用普通的root工具 (root精灵、一键root等) 成功root
    - 后来: 不容易
      - 多数手机想要解锁, 都要官网申请, 通过后, 才能继续root, 否则无法root
        - 比如: 小米的
    - 现在: 也不容易
      - 一般的手机, 都不给root, 也是要申请root才可以
        - 但是好像据说有些手机, 已经被破解了, 淘宝上可以花钱找人在线root
          - 猜测: 估计就是在线解锁BL, 然后继续root的?

## 结论

- 现在如何root
  - 如果本身要破解的安卓app, 能正常安装和运行在安卓模拟器: 那么可以考虑用安卓模拟器, 比如网易的mumu、夜神Nox等
  - 如果只能用真机:
    - 去淘宝上买个别人root好的二手安卓手机
    - 或者: 自己买新的安卓手机, 自己想办法搞定 BL解锁 , 然后再 root

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: 2022-10-30 17:41:50

## 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: 2022-10-30 15:33:31

## 参考资料

- 【已解决】重新给Android13的Pixel5去用Magisk去root
- 【记录】Pixel5中下载安装Magisk最新版v26.1
- 【已解决】给Android13的Pixel5去解锁Bootloader
- 【已解决】Android13的Pixel5中用adb刷入Magisk打了patch后的boot.img以实现root
- 【已解决】Android13的Pixel5中用adb临时启动patch后的boot.img
- 【已解决】Android13的Pixel5中用Magisk永久写入patch后的boot.img
- 【未解决】给Android13的Pixel5刷入用Magisk去Patch后的boot.img
- 【未解决】用Magisk给Pixel5去root重启报错：Android Recovery Cannot load Android system. Your data may be corrupt
- 【已解决】安卓手机Pixel5变砖无法启动系统卡死在Fastboot Mode
- 【已解决】尝试修复Pixel5卡死在Fastboot Mode：Android Flash Tool即flash.android.com
- 
- [打开“共享文件夹”提示没有ROOT权限MuMu模拟器安卓模拟器](#)
- [Root开关功能\\_夜神安卓模拟器新手帮助页](#)
- [Root \(Android\) - 维基百科，自由的百科全书](#)
- [安卓手机的Root是什么意思 - 知乎](#)
- [手机Root是什么意思，安卓手机Root权限获取教程 - 数据蛙](#)
- [什么是ROOT，我们需要它做什么-ZOL手机百科](#)
- [Installation | Magisk](#)
- 

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： 2024-11-14 10:57:01