youpk移植

一款基于ART的主动调用的脱壳机

看雪发布

源码地址： https://github.com/youlor/unpacker

作者说支持pixel 1代且需要刷机。

1. 仅支持pixel 1代
2. 重启至bootloader: `adb reboot bootloader`
3. 解压 Youpk_sailfish.zip 并双击 `flash-all.bat`

百度云：https://pan.baidu.com/s/1ySSy2vNW5TyFjH1LNAcd5w 提取码：vseh

奈何手上只有nexus 5x，故而想移植到nexus 5x。

源码编译

参考文章：为nexus 5x编译android n固件

下载aosp源码

选择分支

这里需要选择7.1.2的系统，因为youpk是基于7.1.2_r33修改的。

针对每款nexus手机的固件，aosp中都有对应的tag，我们需要将源码切换到对应的branch或者tag才可以。
参照https://source.android.com/source/build-numbers ，

OPR6.170623.010	android-8.0.0_r1	Oreo	Pixel C
NZH54D	android-7.1.2_r33	Nougat	Pixel XL、Pixel
NKG47S	android-7.1.2_r32	Nougat	Pixel XL、Pixel
NHG47Q	android-7.1.2_r30	Nougat	Pixel XL、Pixel
NJH47F	android-7.1.2_r29	Nougat	Pixel XL、Pixel
N2G48C	android-7.1.2_r28	Nougat	Nexus 5X、Nexus 6P、Nexus Player、Pixel C
NZH54B	android-7.1.2_r27	Nougat	Pixel XL、Pixel
NKG47M	android-7.1.2_r25	Nougat	Pixel XL、Pixel
NJH47D	android-7.1.2_r24	Nougat	Pixel XL、Pixel
NHG47O	android-7.1.2_r23	Nougat	Pixel XL、Pixel
N2G48B	android-7.1.2_r19	Nougat	Nexus 6P、Nexus Player、Pixel C
N2G47Z	android-7.1.2_r18	Nougat	Nexus 5X

这里选择的版本是N2G48C,，对应的分支为android-7.1.2_r28。也就是说可以将aosp源码切换到这个分支。

源码下载

本次源码是从清华源下载的，参考清华源的Android 镜像使用帮助

这里采用的是初始化包的方案。

下载repo

```
mkdir ~/bin
PATH=~/bin:$PATH
curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
chmod a+x ~/bin/repo
```

使用初始化包

```
wget -c https://mirrors.tuna.tsinghua.edu.cn/aosp-monthly/aosp-latest.tar # 下载初始化包
tar xf aosp-latest.tar
cd AOSP   # 解压得到的 AOSP 工程目录
# 这时 ls 的话什么也看不到，因为只有一个隐藏的 .repo 目录
repo sync # 正常同步一遍即可得到完整目录
# 或 repo sync -l 仅checkout代码
```

此后，每次只需运行 `repo sync` 即可保持同步。

切换分支

```
cd .repo/manifests
// 查看可切换的分支
git branch -a | cut -d / -f 3
// 查看是否包含android-7.1.2_r28分支(为什么要查看呢，因为我的aosp源码是很早之前下载的，如果是新下载的就不用担心啦)
git branch -a | cut -d / -f 3 |grep android-7.1.2_r28

// 切换到android-7.1.2_r28分支
repo init -b android-7.1.2_r28
repo sync
repo start android-7.1.2_r28 --all

// 查看当前的分支
repo branches
```

下载驱动

aosp源码中并不包含厂商的闭源驱动，需要手动下载，地址为 https://developers.google.com/android/drivers

Nexus 5X binaries for Android 7.1.2 (N2G48C)

Hardware Component	Company	Download	SHA-256 Checksum
Vendor image	LG	Link	946f19aea17a9eb58a9af25b24b57f7ace096f3ada244d95cf80262728b4863d
GPS, Audio, Camera, Gestures, Graphics, DRM, Video, Sensors	Qualcomm	Link	658a35dab14a8f0bb3d32cc8fcfd1afc35582e0cab996fef391f6c6d79eeab51

下载N2G48C对应的驱动。

下载解压后均为.sh文件，放到aosp根目录执行脚本即可，中间会要求输入I ACCEPT同意协议。

基础环境

ubuntu18.04

openjdk8

依赖

```
sudo apt-get install libx11-dev:i386 libreadline6-dev:i386 libgl1-mesa-dev g++-
multilib
sudo apt-get install -y git flex bison gperf build-essential libncurses5-
dev:i386
sudo apt-get install tofrodos python-markdown libxml2-utils xsltproc zlib1g-
dev:i386
sudo apt-get install dpkg-dev libsdl1.2-dev libesd0-dev
sudo apt-get install git-core gnupg flex bison gperf build-essential
sudo apt-get install zip curl zlib1g-dev gcc-multilib g++-multilib
sudo apt-get install libc6-dev-i386
sudo apt-get install lib32ncurses5-dev x11proto-core-dev libx11-dev
sudo apt-get install libgl1-mesa-dev libxml2-utils xsltproc unzip m4
sudo apt-get install lib32z-dev ccache
```

编译

```
source build/envsetup.sh
lunch
// 选择aosp-bullhead-userdebug
```

发现此时报错

```
** Don't have a product spec for: 'aosp_bullhead'
** Do you have the right repo manifest?
```

在网上搜了一圈都没找到解决方法，后来在执行make clean的时候发现make出现问题，原来之前为了编译华为手机的内核把make的版本降到了3.8.1，将make更新到4.1就能成功执行lunch了。

继续愉快的编译

```
make -j8
```

错误1

刚开始编译一会就出现

```
build/core/ninja.mk:148:recipe for target 'ninja_wrapper' failed
```

解决方法：在make之前先执行`export LC_ALL=C`

错误2

```
SSL error when connecting to the Jack server. Try 'jack-diagnose'
```

解决方法：网上比较多的是端口占用以及config.properties权限问题，但我这边不是

本次操作如下：

编辑/etc/java-8-openjdk/security/java.security

找到TLSv1这行，

```
jdk.tls.disabledAlgorithms=SSLv3, TLSv1, TLSv1.1, RC4, DES, MD5withRSA, \
```

把TLSv1, TLSv1.1,删除后保存

之后重启jack-admin或者直接重启电脑

参考[JACK错误 SSL error when connecting to the Jack server. Try 'jack-diagnose'](https://www.cnblogs.com/goolinli/p/14793289.html)

之后又能愉快的编译了。

```
### make completed successfully (01:03:41 (hh:mm:ss)) ###
```

看到这个就表示编译成功啦。

刷机

进入源码目录下/out/target/product/bullhead，执行以下命令：

```
//在关机状态下按住电源键和音量减键，进入fastboot模式
//解bl锁，前提是在开发者选项中开启oem解锁，如果手机是5.0以及之前的执行fastboot oem unlock
fastboot flashing unlock

//开始刷机
//指定img镜像位置
export ANDROID_PRODUCT_OUT=out/target/product/bullhead

fastboot -w flashall
```

刷机后会自动重启。

开机后出现如下提示，但不影响使用。**开机过程中系统会检测/system/build.prop 和 /vendor/build.prop，如果发现不一致的地方，就会提示报错**

```
您的设备内部出现了问题。请联系您的设备制造商。
```

移植

从https://github.com/youlor/unpacker下载代码

错误1

```
unkown package name of class file cn/youlor/Unpakcer.class
```

解决方法：

在build/core/tasks/check_boot_jars/package_whitelist.txt文件的`# framework.jar`的下面增加`cn\.youlor`

编译完成后重新刷机

测试

将应用报名写入/data/local/tmp/unpacker.config

```
adb shell "echo com.tencent.mobileqq >> /data/local/tmp/unpacker.config"
```

启动app

查看/data/data/com.tencent.mobileqq/目录下是否有unpacker目录，也可以adb shell logcat |grep unpacker查看脱壳情况。

问题点：$UNPACK_INTERVAL$为10秒是有问题的，如果该app有些检测的话，10秒很可能已经退出了

调用修复工具 dexfixer.jar, 两个参数, 第一个为dump文件目录(必须为有效路径), 第二个为重组后的DEX目录(不存在将会创建)

```
java -jar dexfixer.jar /path/to/unpacker /path/to/output
```

对抗

由于上面编译的userdebug版本，很容易被检测出来。个人比较推荐user版本+MagiskHide功能。

编译user版本

Android主要可以3个编译选项：

eng：工程版本
user：发行版本
userdebug：部分调试版本

这3个选项主要区别(https://source.android.com/source/add-device.html#build-variants)

以下是当前已定义的编译类型：

`eng`	这是默认的编译类型。 • 安装带有 `eng` 和/或 `debug` 标记的模块。 • 除了带有标记的模块之外，还会根据产品定义文件安装相应模块。 • `ro.secure=0` • `ro.debuggable=1` • `ro.kernel.android.checkjni=1` • `adb` 默认处于启用状态。
`user`	这是旨在用作最终版本配置步骤的编译类型。 • 安装带有 `user` 标记的模块。 • 除了带有标记的模块之外，还会根据产品定义文件安装相应模块。 • `ro.secure=1` • `ro.debuggable=0` • `adb` 默认处于停用状态。
`userdebug`	除了以下几点之外，其余均与 `user` 相同： • 还会安装带有 `debug` 标记的模块。 • `ro.debuggable=1` • `adb` 默认处于启用状态。

之前编译的是userdebug版本,本次编译user版本。

在device/lge/bullhead/vendorsetup.sh增加`add_lunch_combo aosp_bullhead-user`,如果想编译eng版本的话可以增加`add_lunch_combo aosp_bullhead-eng`，在执行lunch的时候就可以看到aosp_bullhead-user和aosp_bullhead-eng选项了。

编译好user版本：

链接：https://pan.baidu.com/s/1ROSHqz7Vb6V7Bx6EhO5rxg
提取码：1eu1

ROOT

安装Magisk

https://github.com/topjohnwu/Magisk/releases

下载最新的Magisk-v23.0.apk，将Magisk-v23.0.apk改名为Magisk-v23.0.zip，放到入手机存储中，

```
adb push Magisk-v23.0.zip /sdcard/
```

刷入TWRP

https://dl.twrp.me/bullhead/

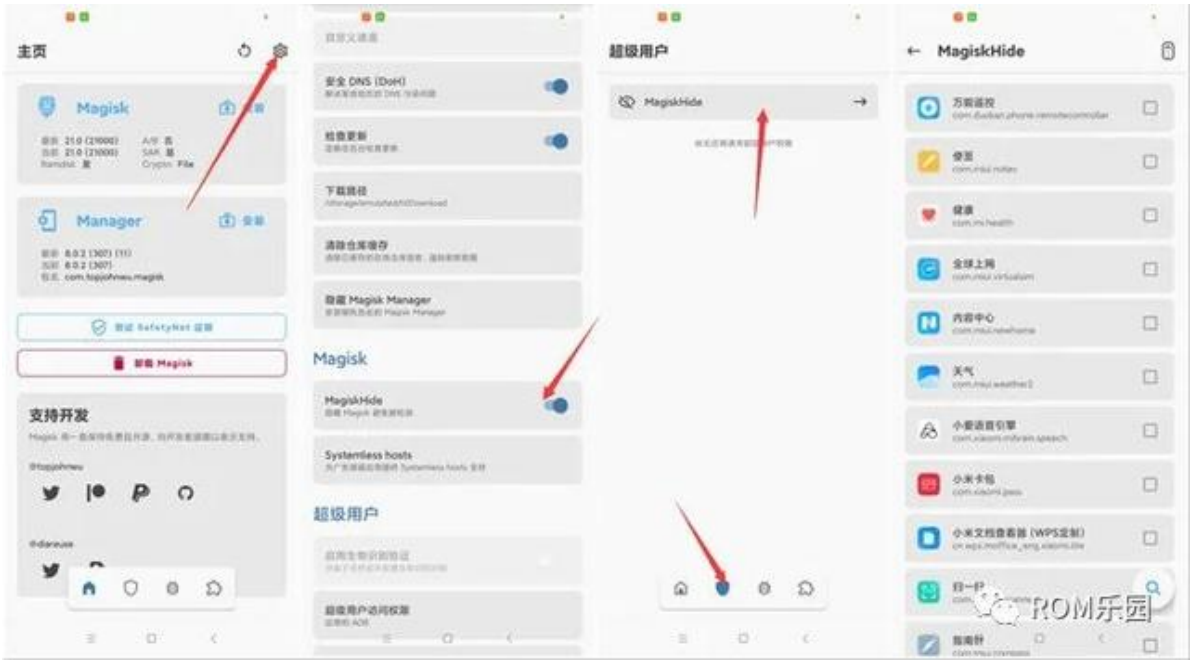
下载最新的twrp-3.5.2_9-0-bullhead.img，手机重启到fastboot模式，我这里是选择刷入了twrp，也可以直接boot，

```
fastboot flash recovery twrp-3.5.2_9-0-bullhead.img
```

在twrp启动后在install中选择Magisk-v23.0.zip进行安装即可。

MagiskHide

在Magisk设置中打开MagiskHide开关，在Magisk中勾选需要对其隐藏的应用。



如果实在过不了检测的话，可以只编译user版本不刷Magisk，将youpk保存的dex路径修改成手机内置存储这些可以直接访问的目录，前提是该应用有访问内置存储的权限，这样youpk还是可以正常脱壳的。修改点在art/runtime/unpacker/unpacker.cc的getDumpDir的返回值，修改之后只需要刷入system.img就行