

xaga内核编译记录

□ 2023-04-07 □ 默认分类 □ 0

前言

在此记录一下编译xaga-s-oss内核的过程，主要目的是为了获取 `vmlinux.h` 用于ebpf程序
之前参考他人的内核编译步骤，总是有这样那样的问题，现在有很多workflow脚本作为参考，过程也是一目了然了

- https://github.com/xiaoleGun/KernelSU_Action/blob/main/.github/workflows/build-kernel.yml
- <https://github.com/tiann/KernelSU/blob/main/.github/workflows/gki-kernel.yml>

步骤

`xaga-s-oss` 即 `Redmi Note11T Pro`，众所周知米系内核编译方法和谷歌官方的略有不一样
所以这里主要参考[KernelSU_Action](https://github.com/xiaoleGun/KernelSU_Action)的处理流程

我的设备是：**VMware + Ubuntu 20.04.4 LTS x86_64**

1. 准备环境

一些必要的库：

```
sudo apt update
sudo apt install git-core gnupg flex bison build-essential zip curl zli
```

2. clang & gcc

通常，我们通过 `build.config.common` 可以知道当前内核编译时clang用的什么版本，比如这里是 `clang-r416183b`

- https://github.com/MiCode/Xiaomi_Kernel_OpenSource/blob/xaga-s-oss/build.config.common

我们最好是下载和这个配置匹配的clang，`build.config.common` 中也有 `BRANCH=android12-5.10` 这样的内容，可以知道其主分支是什么

还有个方法，如果有这个手机，可以知道系统出厂的时候是Android 12，内核版本是5.10，那么在下面的页面中，找到 `android12-5.10` 这个分支

- <https://android.googlesource.com/kernel/common/+refs>

然后查看对应的 `build.config.common` 也能知道对应的clang版本了

- <https://android.googlesource.com/kernel/common/+refs/heads/android12-5.10/build.config.common>
-

另外，如果只有内核源码想知道内核具体版本，可以查看[xaga-s-oss/Makefile](#)，可以知道内核版本是 `5.10.66`

参考[build-kernel.yml](#)，拼接链接，现在可以下载clang了

- <https://android.googlesource.com/platform/prebuilts/clang/host/linux-x86/+archive/refs/heads/master-kernel-build-2022/clang-r416183b.tar.gz>

Q: 为什么要这么下载不直接下载build-tools呢？

A: 因为build-tools里面的通常包括几个版本的clang，属实浪费时间。

gcc这里就选择Android 12最新的分支即可，链接如下：

- https://android.googlesource.com/platform/prebuilts/gcc/linux-x86/aarch64/aarch64-linux-android-4.9/+archive/refs/tags/android-12.1.0_r27.tar.gz

3. 代码同步

如果网不好，下面的步骤自行灵活处理：

```
git clone --depth=1 https://github.com/MiCode/Xiaomi_Kernel_OpenSource.  
cd xaga-s-oss  
mkdir clang-aosp  
wget https://android.googlesource.com/platform/prebuilts/clang/host/lin
```

```
tar -C clang-aosp/ -zxvf clang-r416183b.tar.gz
mkdir gcc-aosp
wget https://android.googlesource.com/platform/prebuilts/gcc/linux-x86/
tar -C gcc-aosp/ -zxvf android-12.1.0_r27.tar.gz
```

4. 正式编译

前面已经把代码和编译工具准备好了，现在可以准备编译了

简单来说就是设置好clang的环境变量，然后指定 `CROSS_COMPILE` 为 `gcc` 的路径，具体如下（完整路径自行更改）：

```
export PATH=/home/kali/Desktop/xaga-s-oss/clang-aosp/clang-r416183b/bin
make -j$(nproc --all) O=out ARCH=arm64 CLANG_TRIPLE=aarch64-linux-gnu-
make -j$(nproc --all) O=out ARCH=arm64 CLANG_TRIPLE=aarch64-linux-gnu-
```

这里必须得指定下 `LD=ld.lld`，不然会出错，具体不知

在执行命令前还需要提醒一下，`xaga-s-oss` 没有 `drivers/mihw` 对应的源代码，所以需要把下面两个文件中的对应配置注释掉：

- `drivers/Kconfig`
- `drivers/Makefile`

我这里是為了编译获取vmlinux，所以这里去掉也无所谓，如果是真的要刷机的，建议从其他分支找到 `mihw` 的源码，看看能不能正常用...

这些处理完了之后就能正常编译内核了，如果重新编译有什么问题，记得 `make clean`

其他：

为了获取vmlinux.h，需要在 `defconfig` 中加入下面两项，再重新编译，当然我这里用的是 `gki_defconfig`，具体路径是 `arch/arm64/configs/gki_defconfig`

```
CONFIG_BPF=y
CONFIG_DEBUG_INFO_BTf=y
```

等等，这里还有一个坑，那就是生成BTf相关的东西的时候还依赖一个库，所以要提前安装好

```
sudo apt install libelf-dev
```

5. 生成vmlinux.h

这里去[bpftool](#)选择最新的action构建，下载 `bpftool`，这里选 `bpftool_amd64`

然后使用下面的命令即可生成 `vmlinux.h`

```
./bpftool btf dump file ./out/vmlinux format c > vmlinux.h
```

□ eBPF 内核