Android Wi-Fi P2P修改安全认证方案操作说明

# 主要步骤

1. 下载Android系统源码，准备编译环境
2. 修改源码，编译framework层代码
3. 使用编译的framework替换原有的框架层jar包

# 准备编译环境

## 1.1配置宿主机系统

参考链接：[Initializing a Build Environment](http://source.android.com/source/initializing.html)

### 1.1.1环境说明

宿主机系统：

1. Android Version >= 2.3.x 64-bit Operating System

>= 5.x(Lollipop) – **Java7**

2.3.x(Gingerbread) ~ 4.4.x(Kitkat) – **Java 6**

2. Android Version < 2.3.x 32-bit OS

1.5(Cupcake) ~ 2.2.x(Froyo) – Java 5

eg：编译Android4.3 or Android4.4.4

Ubuntu14.04 64-bit Operating System

Java 6

PS：如果编译整个Android系统，硬盘空间最好>=150G，如果只部分编译，如果编译framework层代码，那么空间需求比较小。

### 1.1.2安装JDK

参考链接：[在Ubuntu中使用PPA进行Java的多版本安装管理](http://www.linuxidc.com/Linux/2014-03/98691.htm)（支持Ubuntu10.04-Ubuntu14.04）

$sudo add-apt-repository ppa:webupd8team/java //添加软件仓库地址

$sudo apt-get update

$sudo apt-get install oracle-java8-installer //安装Java 8

$sudo apt-get install oracle-java7-installer //安装Java 7

$sudo apt-get install oracle-java6-installer //安装Java 6

$ sudo apt-get install oracle-java6-set-default //以Java 6为例设置环境变量

确定Java版本是否正确：

$java –version //查看java版本

$javac –version //查看javac版本

//版本切换为Java 7

sudo update-java-alternatives -s java-7-oracle

PS:

1. 本质上以上方法使用update-alternatives工具进行多版本软件的管理；

使用以上方法安装好多个版本的java后可以使用如下命令进行版本切换:

$sudo update-alternatives - - config java

$sudo update-alternatives - - config javac

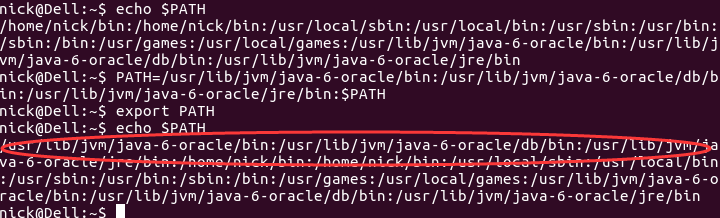
1. 如果以上工具失效（在运行$source ./build/envsetup.sh之后可能导致以上的版本管理方法全部失效，JDK变为最新版本），可以直接在环境变量$PATH中添加对应版本Java的可执行软件路径；

$ PATH=/usr/lib/jvm/java-6-oracle/bin:/usr/lib/jvm/java-6-oracle/db/bin:/usr/lib/jvm/java-6-oracle/jre/bin:$PATH

$export PATH

查看环境变量内容，路径是否添加成功：

$echo $PATH



1. 最简单的是只安装一个版本的Java，但是毕竟版本6太旧了，一般系统都默认安装新版本的JDK

### 1.1.3安装相关依赖包

在Ubuntu14.04 64-bit系统上：

$ sudo apt-get install bison g++-multilib git gperf libxml2-utils make zlib1g-dev:i386 zip

编译系统时如果出现如下错误：

BEGIN failed--compilation aborted at external/webkit/Source/WebCore/make-hash-tools.pl line 23.

make: \*\*\* [out/target/product/generic/obj/STATIC\_LIBRARIES/libwebcore\_intermediates/Source/WebCore/platform/ColorData.cpp] Error 2

target Generated: libwebcore <= external/webkit/Source/WebCore/html/parser/HTMLEntityNames.in

可能还会需要安装：

$ sudo apt-get install libswitch-perl

### 1.1.4配置USB连接权限

新建文件**/etc/udev/rules.d/51-android.rules**，内容格式如下（并且将OWNER="<username>"中的**username**更改为当前Ubuntu系统的用户名）：

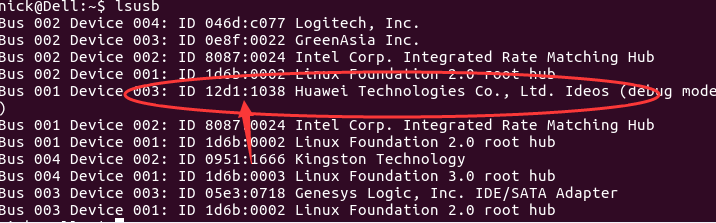
$sudo touch /etc/udev/rules.d/51-android.rules

$sudo chmod a+r /etc/udev/rules.d/51-android.rules

# adb protocol on grouper/tilapia (Nexus 7)  
SUBSYSTEM=="usb", ATTR{idVendor}=="18d1", ATTR{idProduct}=="4e42", MODE="0600", OWNER="<username>"  
#

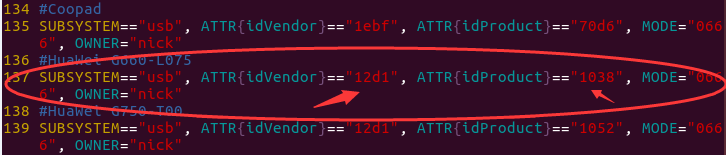
将Android设备通过USB连接到编译系统之后，使用 **$lsusb** 命令查看设备相关属性，然后新建条目，添加对应设备的配置信息。

eg: Huawei G660



注意：ID字段后面的两个字段，这两个字段内容需要填写到51-android.rules文件中，分别对应ATTR{idVendor}，

ATTR{idProduct}属性。



运行命令使得刚才添加的配置生效：

$ sudo service udev restart

### 1.1.5下载Android源码

参考链接：[Downloading the Source](http://source.android.com/source/downloading.html)

1. 安装repo

$ mkdir ~/bin

$ PATH=~/bin:$PATH

$ curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin/repo

$ chmod a+x ~/bin/repo

1. 添加工作目录（最好根据Android版本命名）

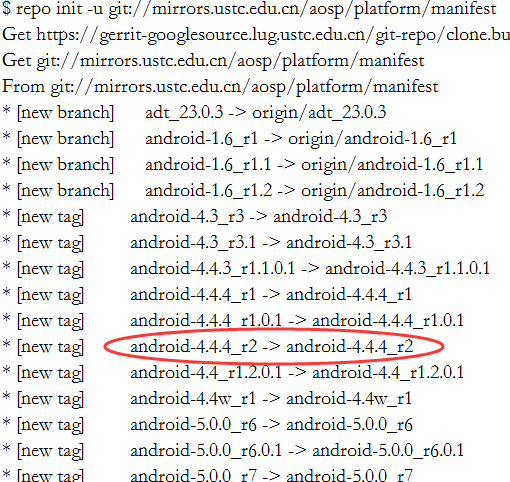
$ mkdir WORKING\_DIRECTORY

$ cd WORKING\_DIRECTORY

1. 选择和对应版本的Android源码

$ repo init -u https://android.googlesource.com/platform/manifest

运行该命令之后会显示一系列Android版本名称，eg：



根据需要加上-b参数选择对应的版本，例如选择Android4.4.4版本，可运行如下命令：

$ repo init -u https://android.googlesource.com/platform/manifest -b android-4.4.4\_r2

备注：Android源码版本一定要与设备上正在运行的Android版本相对应。

1. 下载系统源码

$ repo sync

备注：代码文件比较大，通常>=4.3版本的一般>=25G，根据网速需要耐心等待，如果中途退出，需要反复运行该命令

PS：

1.下载完代码之后如下想查阅该代码对应的Android版本可以通过version\_defaults.mk文件中的PLATFORM\_VERSION变量，eg：

build/core/version\_defaults.mk

PLATFORM\_VERSION := 4.4.4

2.编译整个系统：

参考链接：[Building the System](http://source.android.com/source/building-running.html)

$ source build/envsetup.sh

$ lunch aosp\_arm-eng （或者$ lunch，根据需要选择对应的目标平台）

$ make -j4 （根据CPU核心数和可提供的线程数修改编译线程参数）

如果要加快系统编译速度可以给编译器配置缓存：

$ export USE\_CCACHE=1

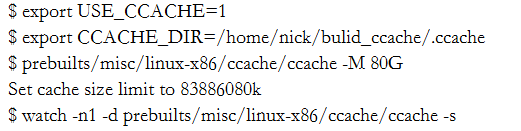
$ export CCACHE\_DIR=~/<path\_of\_your\_choice>/.ccache

$ prebuilts/misc/linux-x86/ccache/ccache -M 50G （建议cache大小为50-100G）

查看编译缓存配置信息：

$ watch -n1 -d prebuilts/misc/linux-x86/ccache/ccache –s

eg：



# 修改和编译framework代码

2.1修改源码

需要修改的源码文件名为WifiP2pService.java，文件路径：

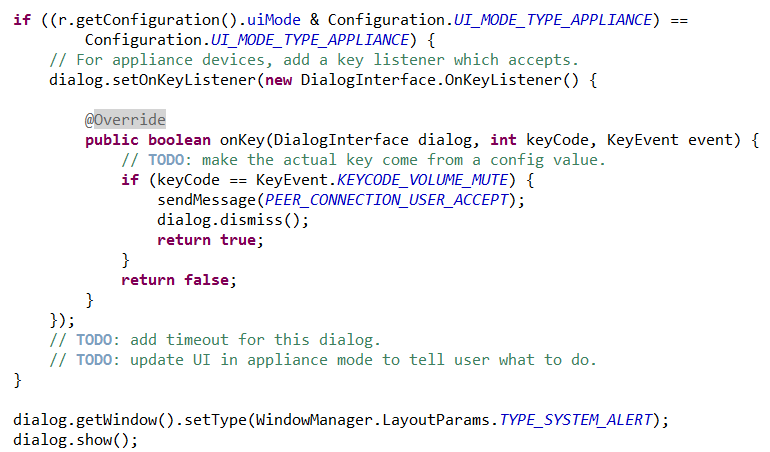
WORKING\_DIRECTORY /frameworks/base/wifi/java/android/net/wifi/p2p

需要修改的源码位于函数 **private** **void** notifyInvitationReceived() 中：

1. 完全注释AlertDialog dialog变量声明代码段：

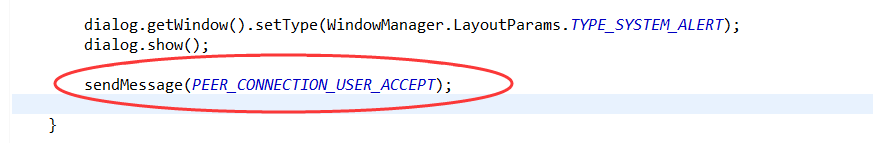


1. 注释掉一部分逻辑判断和dialog显示代码：



1. 添加发送认证通过的消息代码语句

在上一段注释掉的代码后面添加语句sendMessage(*PEER\_CONNECTION\_USER\_ACCEPT*);



备注：经过修改后的代码放在附件中，对应系统版本Android4.3，代码大概位于第2076行附件的位置。

Android4.4.4, 2140.

如此简单粗暴的修改导致在应用层使用Wi-Fi Direct功能时，只能选择PBC安全认证模式，Keypad或者Display模式此时都无法使用了；并且由于系统Settings中的Wi-Fi Direct默认使用PBC模式因此，通过Settings使用Wi-Fi Direct功能时不会出现不正常现象，不过此时Android系统的Wi-Fi Direct功能也就失去安全保障。

另外可以继续仔细修改代码与逻辑，使得PBC模式自动认证时其它模式也可以正常使用，这需要后续的工作。

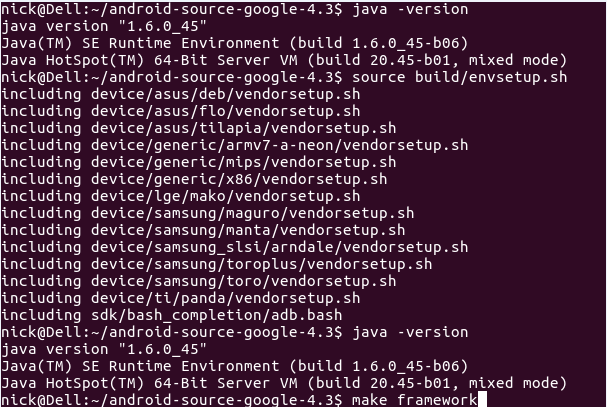
## 2.2编译framework

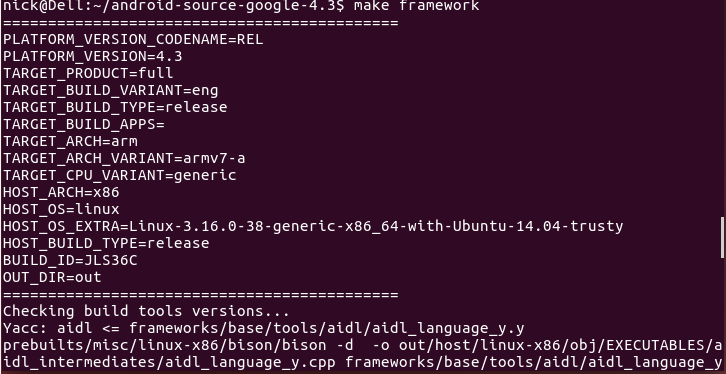
$ cd WORKING\_DIRECTORY

$source build/envsetup.sh

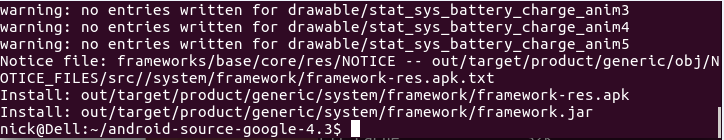
$make framework

eg: Android 4.3





编译结束显示如下信息，生成三个framework层相关文件：framework.jar, framework.odex, framework-res.apk



如果是Android4.4.4系统会生成5个文件，比4.3多出framework2.jar, framework2.odex两个文件。

# framework代码替换

## Android设备开启开发者调试模式

通过设置，进入手机配置信息界面，多次点击版本号：



## Root Android设备

分别以Android原装系统与国产手机为例说明（还有很多其他方法可以通过网络获知）

Nexus 5获取root权限：

$adb reboot bootloader

$fastboot oem unlock

Coolpad 8730L：

1. 安装[Root 精灵](http://www.shuame.com/root/) PC版（Windows操作系统）
2. 连接设备，一键Root



## 安装Android工具: ADB, Fastboot

$ sudo apt-get install android-tools-adb android-tools-fastboot

查看连接的设备信息（使用adb工具时一次连接一个Android设备）：

$ adb devices

List of devices attached

1556f473 device

查看设备详细信息：

$ adb devices -l

List of devices attached

1556f473 device usb:1-1.3 product:Coolpad8730L model:Coolpad\_8730L device:Coolpad8730L

连接设备，获取root权限，添加写权限：

$adb remount

remount failed: Operation not permitted （显示权限问题，需要Root设备）

$ adb shell

shell@Coolpad8

$ su

# mount -o rw,remount -t yaffs2 /dev/block/mtdblock3 /system

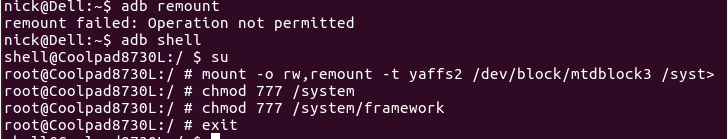
# chmod 777 /system

# chmod 777 /system/framework

# exit

$

eg:



PS:

1）每次使用adb工具进行设备文件的替换都需要运行以上步骤获取写权限。

2）如果adb已经运行并且无法识别设备可以使用命令 **$ adb kill-server** 重启adb和configuration。

## 替换framework的软件包

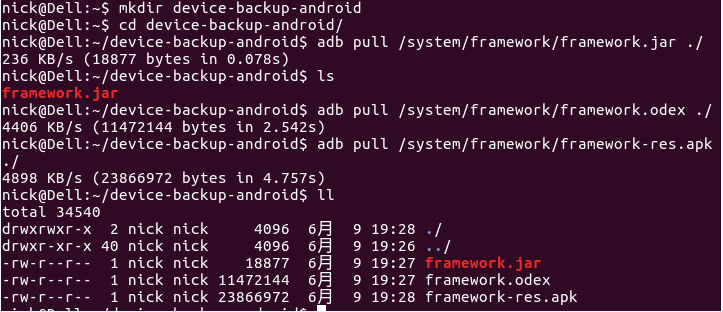
1. 替换之前可以先备份原设备framework软件包

$mkdir backup-directory

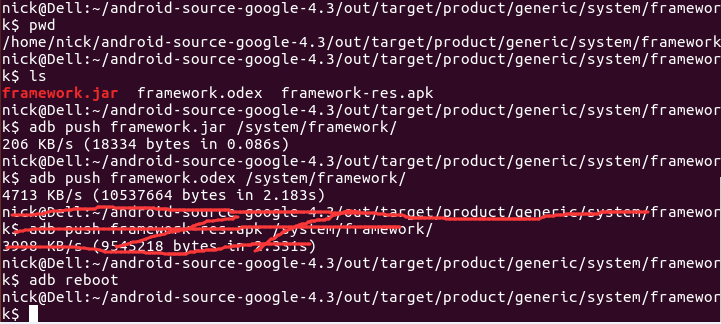
$cd backup-directory

$ adb pull /system/framework/framework.jar ./

$ adb pull /system/framework/framework.odex ./



eg:



备注：framework-res.apk不需要备份和替换，否则设备重新启动时可能发生意想不到问题。

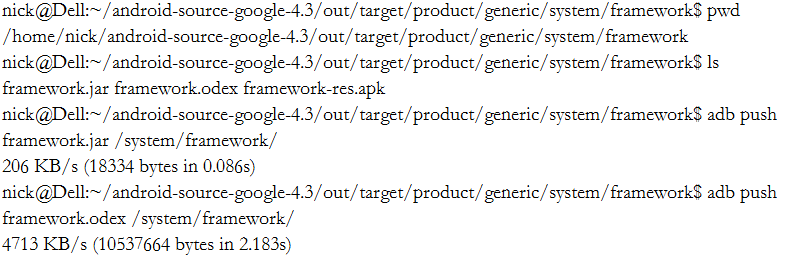
1. 写入自编译的framework软件包

$cd WORKING\_DIRECTORY /out/target/product/generic/system/framework

$ adb push framework.jar /system/framework/

$adb push framework.odex /system/framework/

eg：



备注：其它设备（eg. Nexus 5），除了以上两个文件还可能产生framework2.jar，framework2.odex文件，也需要进行替换。

1. 重启设备

$adb reboot

启动界面可能显示：Android正在升级... 优化应用……

启动完成后，系统运行正常则替换成功。

备注：

最好使用支持Android原生系统的设备进行操作，否则可能由于各个厂商对设备系统的修改产生兼容性问题，因此非Android原生系统设备（如Coolpad 8730L），最好事先进行系统可恢复的准备。例如Recovery是否支持SD卡刷机，如果不支持则先刷一个第三方的Recovery，以免出现问题可以使用其它ROM使系统恢复正常；或者利用fastboot工具采用线刷方式，刷回原版本（例如Coolpad 8730L线刷）。

我使用国产系统Coolpad 8730L替换framework包时，由于也替换了framework-res.apk包，然后出现问题，几乎成砖，后来借用Nexus 5替换和验证成功；至于Coolpad 8730L是否可以进行framework替换还未来得及再次验证。

nick@Dell:~$ adb devices -l

List of devices attached

060ea4a93443b702 device usb:3-3 product:hammerhead model:Nexus\_5 device:hammerhead

