Android程序分析环境搭建-Linux篇

上一篇讲了如何在macOS上搭建Android程序分析环境，现在，我们继续讲解如何在Linux上完成这些操作。

Ubuntu分析环境搭建

Android官方推荐使用Ubuntu操作系统来编译Android系统源码。Ubuntu系统的使用方法简单，界面友好，是最受欢迎的Linux系统之一。下面介绍的Linux平台分析环境搭建，以及Android源码编译环境，使用的Linux系统均是64位的Ubuntu系统，版本为16.04。

安装JDK

Ubuntu系统上安装JDK有两个可选方案：Oracle官方提供的JDK，以及第三方开源的OpenJDK。使用这两个版本的JDK，都能够正常的在生产环境中开发Android软件与编译Android源码。如果读者打算使用Oracle官方的JDK，可以按照下面的步骤来操作。

到Oracle官方网站下载JDK安装包。下载地址为http://www.oracle.com/technetwork/java/javase/downloads/index.html，以JDK的8u131版本为例，将下载的jdk-8u131-linux-x64.tar.gz文件放到任意目录下，本例中解压到~/java目录下，打开一个终端环境输入以下命令对其进行解压：

```
$ tar xvzf jdk-8u131-linux-x64.tar.gz
```

解压完成后，会在当前目录下多出一个jdk1.8.0_131目录。执行以下命令设置`JAVA_HOME`环境变量，并将其bin目录添加到`PATH`环境变量中。

```
$ echo JAVA_HOME=~/java/jdk1.8.0_131 >> ~/.profile
$ export JAVA_HOME >> ~/.profile
$ export PATH=$JAVA_HOME/bin:$PATH >> ~/.profile
$ source ~/.profile
```

最后，执行`java -version`检查是否安装配置成功，如下所示：

```
$ java -version
java version "1.8.0_131"
Java(TM) SE Runtime Environment (build 1.8.0_131-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)
```

安装Oracle JDK的另一种方式，是通过添加PPA命令行方式安装。例如想要在Ubuntu 16.04系统上编译Android 4.4版本的源码，则需要安装JDK 6。执行以下命令可以将Oracle JDK6安装到本机。

```
$ sudo add-apt-repository ppa:webupd8team/java
$ sudo apt-get update
$ sudo apt-get install oracle-java6-installer
```

Android官方采用OpenJDK来编译Android系统源码。下面演示如何通过OpenJDK的方式来安装OpenJDK 7，执行如下命令：

```
$ sudo add-apt-repository ppa:openjdk-r/ppa
$ sudo apt-get update
$ sudo apt-get install openjdk-7-jdk
```

Ubuntu系统上可以同时安装保留多个版本的JDK，在使用时，根据需要可以在不同的版本之间进行切换。执行以下命令可以切换`java`到JDK 7版本：

```
$ sudo update-alternatives --config java
[sudo] password for android:
There are 3 choices for the alternative java (providing /usr/bin/java).

  Selection    Path                                            Priority
Status
------------------------------------------------------------
  0            /usr/lib/jvm/java-6-oracle/jre/bin/java          1082
auto mode
  1            /usr/lib/jvm/java-6-oracle/jre/bin/java          1082
manual mode
* 2            /usr/lib/jvm/java-7-openjdk-amd64/jre/bin/java   1071
manual mode
  3            /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java   1081
manual mode

Press <enter> to keep the current choice[*], or type selection number: 2
```

执行以下命令切换`javac`到OpenJDK 7版本：

```
android@ubuntu1604:~/Downloads$ sudo update-alternatives --config javac
[sudo] password for android:
There are 3 choices for the alternative javac (providing /usr/bin/javac).

  Selection    Path                                            Priority
Status
------------------------------------------------------------
  0            /usr/lib/jvm/java-6-oracle/bin/javac            1082      auto
mode
  1            /usr/lib/jvm/java-6-oracle/bin/javac            1082
manual mode
* 2            /usr/lib/jvm/java-7-openjdk-amd64/bin/javac     1071
manual mode
  3            /usr/lib/jvm/java-8-openjdk-amd64/bin/javac     1081
manual mode

Press <enter> to keep the current choice[*], or type selection number: 2
```

完成上述操作后，执行以下命令查看JDK版本是否切换成功，如下所示，JDK已经成功切换为OpenJDK 7：

```
$ java -version
java version "1.7.0_95"
OpenJDK Runtime Environment (IcedTea 2.6.4) (7u95-2.6.4-3)
OpenJDK 64-Bit Server VM (build 24.95-b01, mixed mode)
$ javac -version
javac 1.7.0_95
```

安装Android SDK

在Ubuntu系统上安装Android SDK，有下载Android SDK包后通过命令行方式更新，以及使用`Android Studio`内置的SDK Manager方式来安装这两种方式。这两种方式的安装方法与macOS系统一样，此处不再赘述。

安装Android NDK

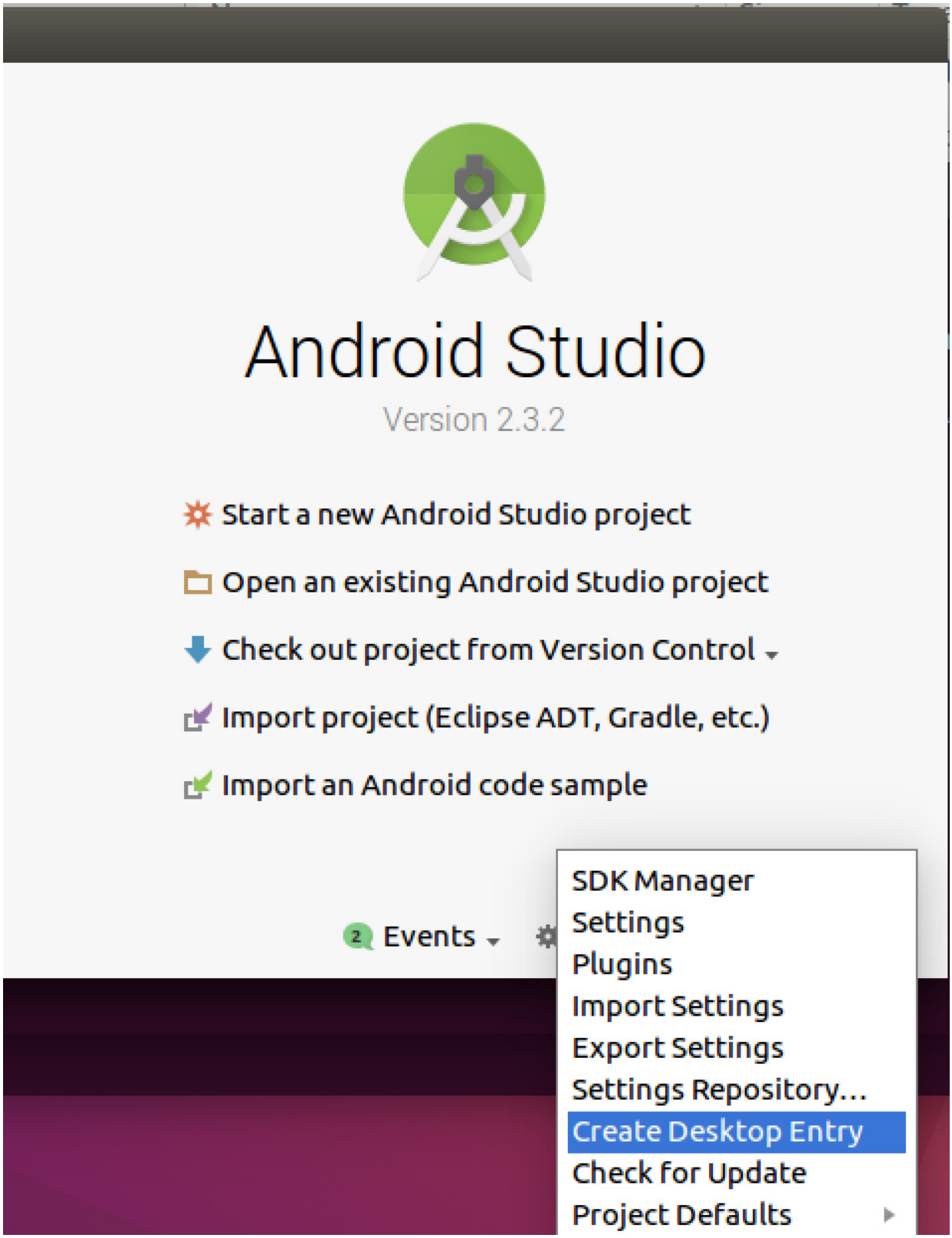
在Ubuntu系统上安装Android NDK，同样有下载Android SDK包后通过命令行方式更新、下载Android NDK压缩包后解压、以及使用`Android Studio`内置的SDK Manager方式来安装这几种方式。这些安装方法与macOS系统上操作是一样的，此处不再赘述。

Android Studio集成开发环境

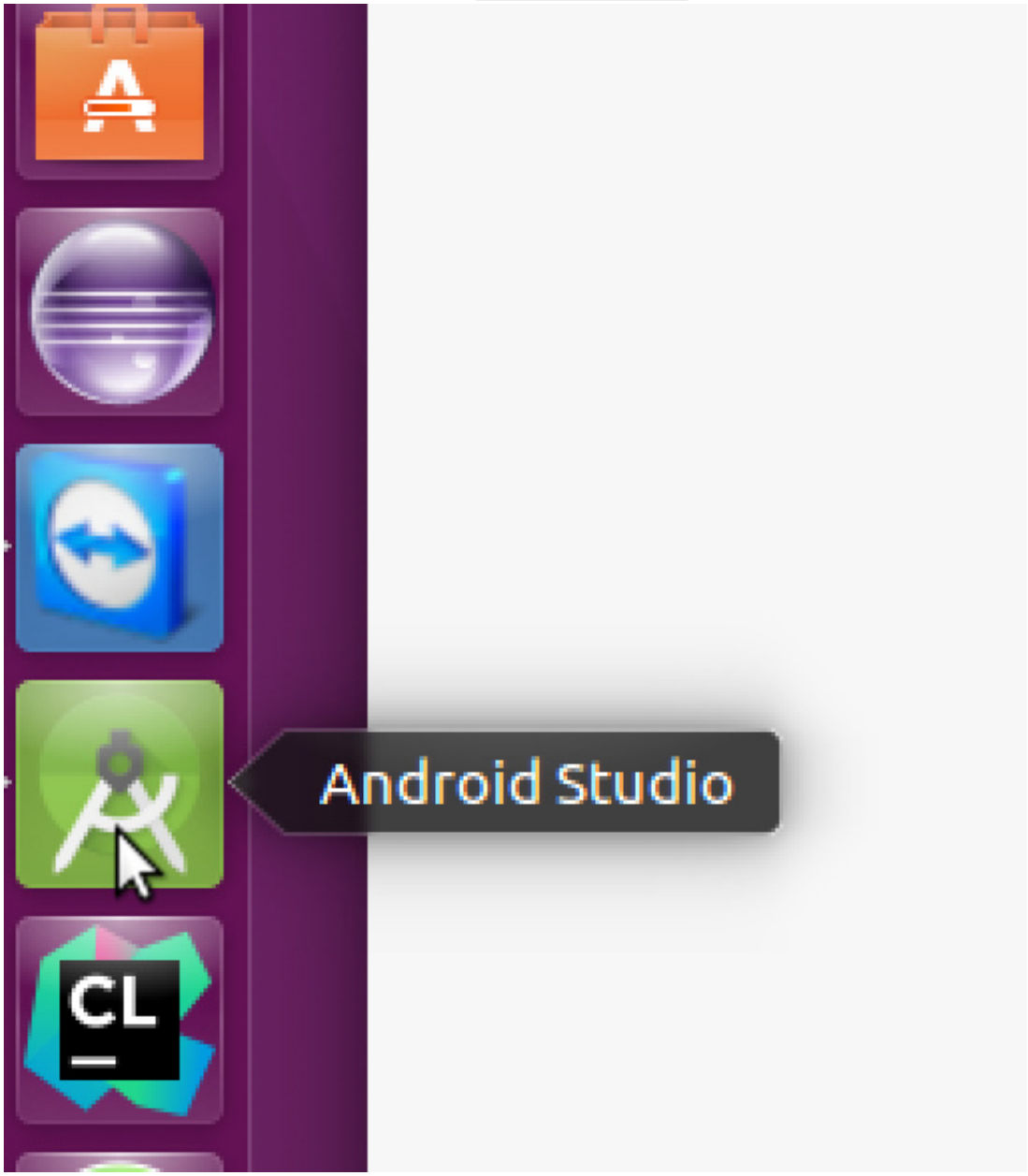
Ubuntu系统上的`Android Studio`以zip压缩包提供，它的启动程序是一个名为studio.sh的脚本。执行以下命令解压并运行`Android Studio`。

```
$ mkdir -p ~/Android
$ unzip ~/Downloads/android-studio-ide-162.3934792-linux.zip -d ~/Android
$ ~/Android/android_studio/bin/studio.sh
```

启动完成后进行主界面，点击界面上的Configure->Create Desktop Entry，创建桌面图标项，如图所示：



创建完成后，可以在系统左侧边栏上看到`Android Studio`的图标，如图所示：



以后就可以通过点击图标来启动`Android Studio`了。启动`Android Studio`完成后，安装Android SDK、安装Android NDK、以及创建Android模拟器的步骤与Windows系统操作是一样的，此处不再展开。

常用逆向分析工具

以下的几款工具，在Android逆向分析领域，使用的最为频繁，几乎是所有安全研员人员必备的工具：

- ApkTool。`ApkTool`是Android逆向分析领域最久远的工具，它提供了Android APK的反编译与回编译功能，这让APK的重打包成为了一种可能，虽然现在越来越多的工具趋向于使用GUI来进行反编译操作，但命令行方式操作的`ApkTool`更加简单与便捷，仍然是很多分析人员的最佳选择。
- Smali/BakSmali。`Smali/BakSmali`是DEX文件的反编译与回编译工具。它提供了一种Smali语法格式，使得DEX文件的修改可以基于Smali的汇编指令来完成。`ApkTool`底层就使用它提供的dexlib库来生成Smali反汇编文件。

- dex2jar。`dex2jar`是使用最多的DEX文件转Jar包的工具。使用它转换出的Jar文件，可以使用`JD-GUI`等工具查看它的Java代码。
- 010 Editor。跨平台的二进制编辑器。它的强大在于可以使用文件格式模板对二进制文件进行查看与编辑。官网提供了丰富的二进制文件格式模板供分析人员下载。在一些逆向分析的场景中，分析人员可以使用`010 Editor`内置的脚本语言，编写二进制文件格式模板进行文件格式分析与测试。可以说，`010 Editor`是分析二进制程序必备的工具之一。

常用的Linux Shell命令

为了更好的展示特定操作的输出反馈信息，以后在讲解分析内容时，会大量的使用了Linux Shell命令，这些命令同时可以在Windows、macOS、Linux等系统上运行，并且在输出结果上保持着高度的一致性。

`ls`命令用于列文件目录。例如在`Cygwin`环境中，查看Windows系统上D盘中的目录，可以执行如下命令：

```
$ ls /cygdrive/d
```

`cat`是一个比较常用的命令，用于在终端中输出文件的内容。例如查看当前目录下build.gradle文件的内容，可以执行下面的命令：

```
$ cat ./build.gradle
```

`grep`命令用来搜索匹配的文本内容，它的使用方法比较丰富。例如在当前目录下，搜索并列出所有包含“hello”字符串的文件，可以执行如下命令：

```
$ grep -r "hello" ./
```

给`grep`传入“-A”参数可以输出指定行数的结果。例如只输出上面结果的前10条信息，可以执行如下命令：

```
$ grep -r "hello" ./ -A 10
```

`grep`命令可以通过管道形式进行多次调用，例如搜索并列出当前目录下，所有同时包含“hello”与“world”字符串的文件，可以执行如下命令：

```
$ grep -r "hello" ./ | grep "world"
```

`export`命令用于导出一个环境变量，在外部可以通过添加一个“
”符号进行访问导出的环境变量。例如将‘as‘汇编器的路径导出为‘AS‘环境变量，并通过调用‘AS`命令生成hello.s的目标文件，可以执行如下命令：

```
$ export AS="/usr/local/opt/android-ndk/toolchains/arm-linux-androideabi-
4.9/prebuilt/darwin-x86_64/bin/arm-linux-androideabi-as"
$ $AS hello.s -o hello.o
```

`mkdir`命令用于创建目录，传入“-p”参数可以一次性创建多级目录。例如，在当前目录下创建foo/bar目录，可以执行如下命令：

```
$ mkdir -p foo/bar
```

编译Android源码

Ubuntu是Android官方默认支持的Android系统源码编译系统，只需要按照官网的环境配置说明进行配置即可。

首先第一步是安装JDK，编译不同版本的Android源码可能需要安装不同版本的JDK，这点在之前已经讲过，此处不再赘述。

然后是安装下载与编译Android编译源码所需的软件，只需要执行以下命令即可：

```
$ sudo apt-get install git-core gnupg flex bison gperf build-essential \
  zip curl zlib1g-dev gcc-multilib g++-multilib libc6-dev-i386 \
  lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z-dev ccache \
  libgl1-mesa-dev libxml2-utils xsltproc unzip
```

安装完成后，按下来是下载`repo`与配置git用户名，执行如下命令：

```
$ curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
$ chmod a+x ~/bin/repo
$ git config --global user.name "Your Name"
$ git config --global user.email "you@example.com"
```

接着是下载与同步源码，执行如下命令：

··· $ repo init -u https://android.googlesource.com/platform/manifest -b android-7.1.1_r1 $ repo sync --force-sync --force-broken ···

下载源码完成后，就是编译工作了，执行如下命令即可：

```
$ export USE_CCACHE=1
$ export CCACHE_DIR=ccache
$ prebuilts/misc/linux-x86/ccache/ccache -M 50G
$ source build/envsetup.sh
$ lunch aosp_angler-userdebug
$ make clobber
$ make -j8
```

小结

本篇主要介绍了在Ubuntu平台上，如何搭建安卓的开发与分析环境，以及如何在Ubuntu系统上编译安卓系统的源码。

更多精彩内容，欢迎关注微信公众号【feicong_sec】