

目录

前言	1.1
Android逆向开发概览	1.2
Android逆向典型流程	1.3
安卓root	1.3.1
安装apk	1.3.2
搞懂逻辑	1.3.3
静态分析	1.3.3.1
动态调试	1.3.3.2
输出结果	1.3.4
重新打包apk	1.3.4.1
相关工具	1.4
adb	1.4.1
相关心得	1.5
文件管理	1.5.1
文件管理器	1.5.1.1
相关资料	1.6
Android API Level	1.6.1
子教程	1.7
附录	1.8
参考资料	1.8.1

Android逆向开发

- 最新版本: v1.1.0
- 更新时间: 20231004

简介

介绍Android逆向开发相关的内容。包括安卓逆向开发的典型流程图和脑图、安卓逆向的核心思路；以及Hook插件与开发，包括Xposed、Cydia Substrate，以及其插件Android-OpenDebug、Introspy-Android；以及其他心得，包括adb、文件管理、文件管理器、相关资料；以及其他相关子教程。

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_reverse_dev: Android逆向开发](#)

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逆向开发 book.crifan.org](#)
- [Android逆向开发 crifan.github.io](#)

离线下载阅读

- [Android逆向开发 PDF](#)
- [Android逆向开发 ePub](#)
- [Android逆向开发 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](mailto:admin@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](#)

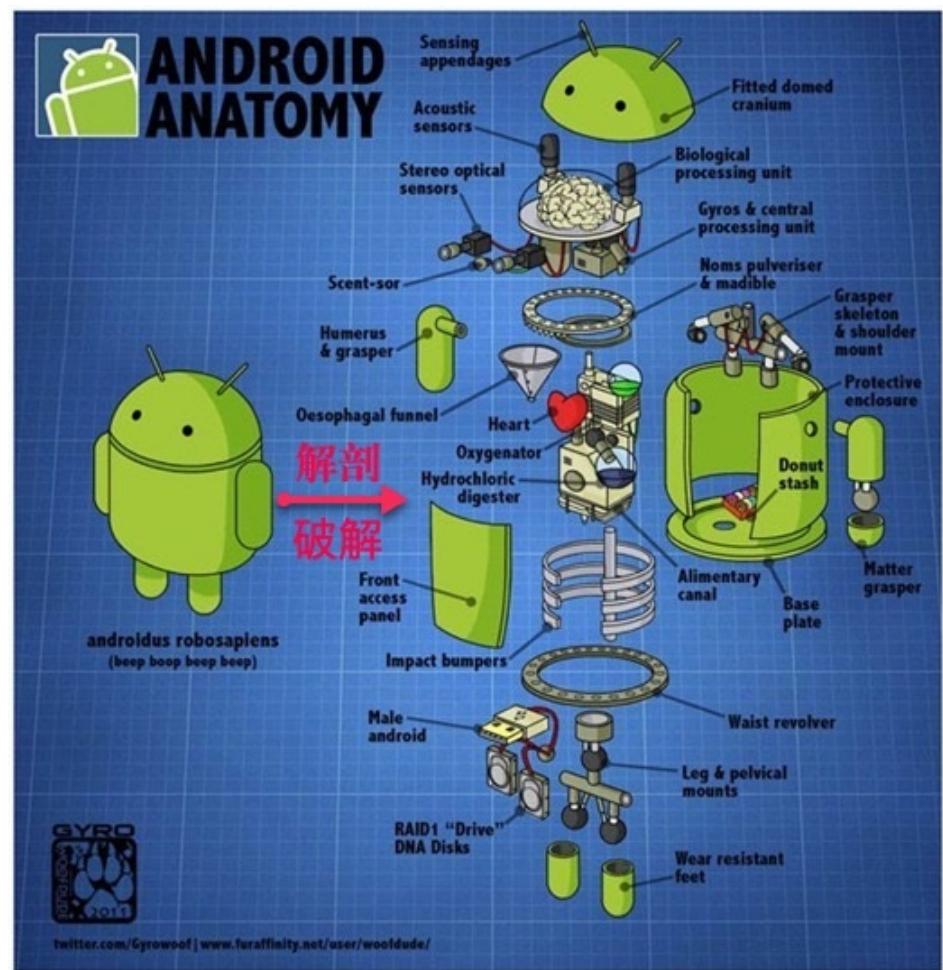
crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：

2023-10-04 16:30:28

Android逆向开发概览

此处介绍[安卓的安全和逆向](#)中的 安卓逆向 中关于 安卓逆向开发 相关的内容：

- 背景
 - 安卓应用的 存在形式
 - 静态的
 - 安装文件: apk
 - 动态的
 - 运行中的程序叫: app
- 安卓逆向开发 = 安卓破解
 - 安卓破解 = android crack
 - 破解
 - = crack = cracking
 - 也叫 安卓逆向工程 = 安卓反向工程
 - 而逆向工程, 就像对一个人去解剖



- 概述
 - 核心流程
 - 安卓root
 - 安装apk
 - 搞懂逻辑

- 静态分析
- 动态调试
- 输出结果
 - 重新打包apk
- 详见
 - [Android逆向典型流程](#)

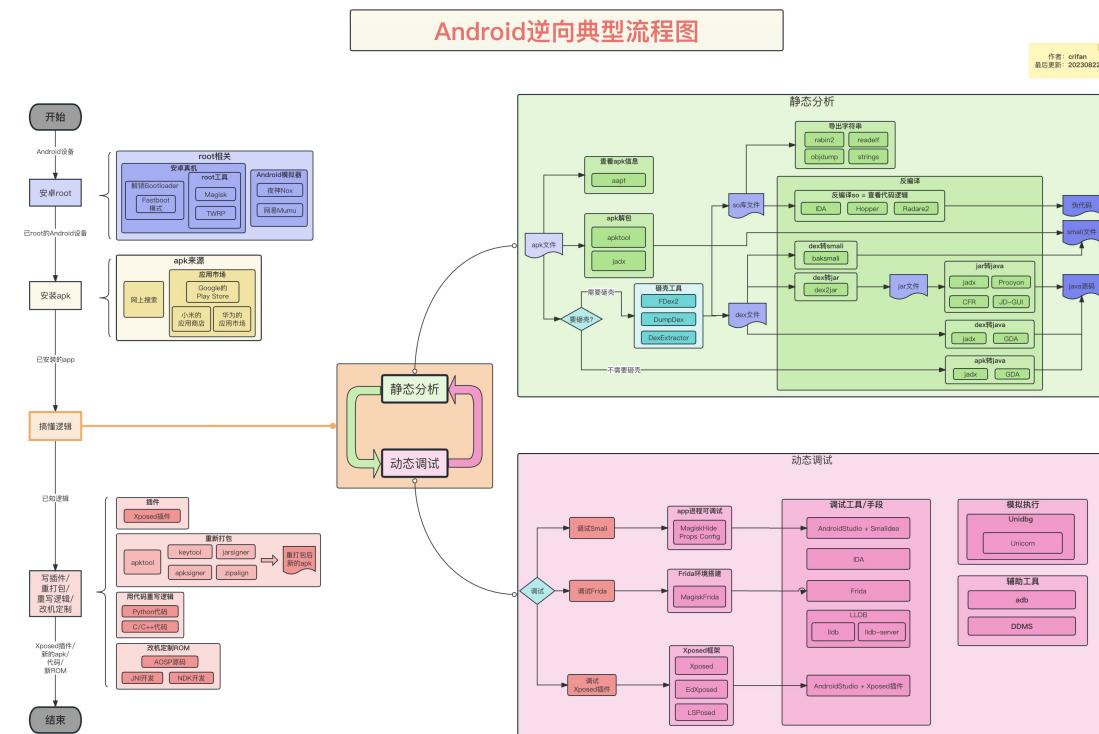
crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新:
2023-08-30 21:10:34

Android逆向典型流程

此处整理出，安卓逆向的典型流程图和脑图：

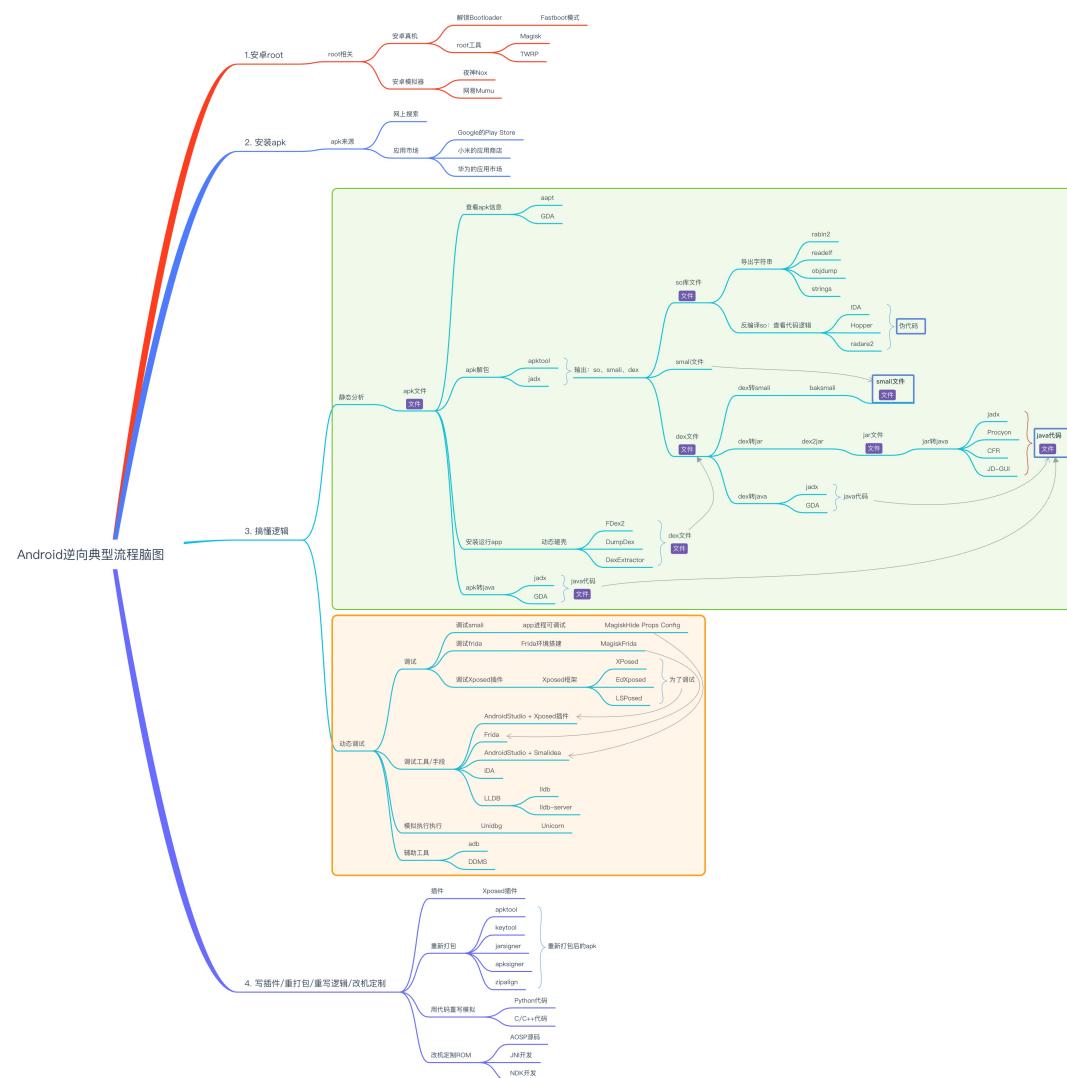
Android逆向典型流程图

- Android逆向典型流程图
 - 在线浏览
 - [Android逆向典型流程图 20230822| ProcessOn免费在线作图](#)
 - 离线查看



Android逆向典型流程脑图

- Android逆向典型流程脑图
 - 在线浏览
 - [Android逆向典型流程脑图 20230821| ProcessOn免费在线作图](#)
 - 离线查看



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:
2023-09-13 21:29:03

安卓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：
2023-08-25 21:09:23

安装apk

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 21:09:54

搞懂逻辑

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 21:10:50

静态分析

此处主要介绍安卓的逆向破解中的静态分析：

- 安卓的逆向破解 静态分析
 - 按文件格式类型分
 - 针对 apk
 - 查看apk信息
 - aapt
 - 解包工具：输出 dex 、 so 、 smali
 - apktool
 - 反编译工具
 - 直接apk转java
 - jadx
 - JEB
 - GDA
 - 针对 dex
 - 各种导出dex的工具=砸壳工具=脱壳工具
 - FDex2
 - DumpDex
 - DexExtractor
 - 各种反编译dex的工具
 - dex转jar
 - dex2jar
 - dex转smali
 - baksmali
 - dex直接转java
 - jadx
 - GDA
 - 针对 jar
 - jar转java=各种反编译工具
 - Procyon
 - CFR
 - JD-GUI
 - 针对 so 库文件=（往往是ARM64架构）的ELF文件
 - 导出静态资源
 - readelf
 - objdump
 - razbin2
 - 反编译代码逻辑
 - IDA
 - Hopper
 - Radare2

- Ghidra

- 涉及子领域
 - 反代码混淆

详见独立子教程：

[Android逆向：静态分析](#)

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:

2023-09-13 21:24:34

动态调试

此处主要介绍安卓逆向破解中的动态调试：

- 安卓的逆向破解 动态调试
 - (绕过限制去) 抓包
 - 绕过证书绑定的Exposed插件或Frida的脚本
 - 反root检测
 - 反反调试
 - 各种动态调试手段
 - 调试smali: AS+smalidea插件
 - app进程可调试
 - Magisk插件: MagiskHide Props Config
 - 调试Frida: Frida
 - Frida环境搭建
 - Magisk插件: MagiskFrida
 - 调试lldb: LLDB
 - 调试Xposed插件
 - XPosed
 - EdXposed
 - LSXposed
 - 模拟代码执行
 - Unidbg
 - Unicorn
 - 辅助调试工具
 - adb
 - DDMS

详见独立子教程：

[Android逆向：动态调试](#)

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：

2023-09-14 20:48:38

输出结果

此处介绍安卓的逆向破解的最终输出结果：

- 安卓的逆向破解 输出
 - 重新打包apk
 - apktool
 - keytool
 - jarsigner 或 apksigner
 - zipalign
 - 用代码重写逻辑
 - Python代码
 - C/C++代码
 - 改机定制ROM
 - AOSP源码 编译
 - JNI 开发
 - NDK 开发

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 21:13:24

重新打包apk

此处介绍安卓逆向中输出结果中的：

- 重新打包apk
 - 概述
 - apktool
 - keytool
 - jarsigner 或 apksigner
 - zipalign
 - 详解
 - [Android逆向：重新打包apk](#)

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 21:17:18

相关工具

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：
2023-08-25 22:08:35

adb

TODO:

- 【整理】adb命令语法帮助信息
- 【已解决】Mac中运行adb shell无需su超级用户即可正常运行输出结果
- 【未解决】尝试解决adb只有su超级用户权限才能工作：adb版本
- 【已解决】Android Studio和安卓设备Google Pixel3重启后adb shell再次异常：只有su超级用户权限才能正常运行命令
- 【已解决】尝试解决adb没有su超级用户权限无法运行子命令：ps进程相关
- 【已解决】Android Studio中对于ADB：Use existing manually managed server且指定adb端口
- 【已解决】Mac中给adb start-server指定端口
- 【已解决】Mac中adb shell子命令运行卡死无响应
- 【已解决】安卓手机adb shell报错：Can't find service
- 【已解决】安卓手机Google Pixel3中adb命令pm运行报错：cmd Can't find service: package
- 【已解决】安卓手机adb shell service list报错：Found 0 services
- 【已解决】安卓手机adb shell无法直接运行命令但是进入shell后su后却可以
- 【已解决】adb shell带su超级用户权限去运行子命令
- 【未解决】pm安装重签名的apk报错：Failure INSTALL_FAILED_UPDATE_INCOMPATIBLE
- 【已解决】pm安装apktool重新打包的apk报错：Failure
INSTALL_PARSE_FAILED_NO_CERTIFICATES
- 【无需解决】安卓手机pm卸载app报错：Failure DELETE_FAILED_INTERNAL_ERROR
- 【已解决】安卓手机中用adb shell中的pm去安装YouTube的apk
- 【已解决】安卓手机Google Pixel3中用pm安装YouTube的apk报错：Failure
INSTALL_FAILED_VERSION_DOWNGRADE

用adb安装apk

- adb install xxx.apk = adb push xxx.apk somePath + pm install /somePath/xxx.apk

举例：

```
crifan@licrifandeMacBook-Pro ~ ~/dev/dev_tool/android/EdXposed$ pwd
/Users/crifan/dev/dev_tool/android/EdXposed
crifan@licrifandeMacBook-Pro ~ ~/dev/dev_tool/android/EdXposed$ ll
total 8224
crifan@licrifandeMacBook-Pro ~ ~/dev/dev_tool/android/EdXposed$ adb install EdXposedManager-4.6.2-46200-org.meowcat.edxposed.manager-release.apk
Performing Streamed Install
Success
```

adb shell

命令提示符

adb shell 进入shell后：

- 命令行提示符
 - # = 并号 : root 用户
 - \$ = 美元 符号: 普通 用户

另外，也可以通过：

```
whoami
```

查看当前用户是什么

举例：

- root用户

```
blueline:/ # whoami
root
```

- 普通用户： shell

```
13 blueline:/ $ whoami
shell
```

adb shell命令行前面的数字

正常情况， shell前面是没有数字的：

```
[ adb shell
blueline:/ # pwd
/
```

但是，如果前面出现一个数字加上竖杠，则表示：前一次命令执行的返回值，前一个命令运行出错了的出错码

比如：

```
blueline:/ # pm --help
cmd: Can't find service: package
20 blueline:/ #
```

此处的 20| 就是 前一个命令运行出错的返回值 = 出错码

而继续运行，如果后续命令正常运行，则出错码就消失了：

```
20 blueline:/ # which pm
/system/bin/pm
blueline:/ #
```

其实表示的是：

- 上一个命令运行结果=返回值
 - 0 : 表示没有出错
 - 所以就不显示出错码
 - 非0 : 就显示，提示你出错了

```
crifan@licrifandeMacBook-Pro ~ ~/dev/dev_tool/android/reverse_engineering/adb_root ➤ adb shell
blueline:/ # pwd
/
blueline:/ # pm --help
cmd: Can't find service: package
20|blueline:/ # which pm
/system/bin/pm
blueline:/ # █
```

adb常见命令

代码	含义
adb reboot bootloader	在 bootloader 模式下重启
adb push	将文件从本地系统复制到 Android 手机的位置
adb pull	将文件从 Android 复制到您的系统
adb devices	显示所有连接的 adb 兼容设备
adb backup	备份 Android 设备
adb install	将应用程序从系统的 apk 文件位置安装到 Android 设备上
adb reboot	在正常模式下重新启动 Android 手机
adb connect	通过 WiFi 网络使用adb命令
adb shell screencap	获取设备的屏幕截图

值) 什么值得买

- 相关
 - 常用 Fastboot 命令

代码	含义
fastboot devices	显示连接的 Android 设备的序列号
fastboot oem unlock	解开 bootloader 锁 (Android 5.0 及以下)
fastboot oem lock	恢复 bootloader 锁 (Android 5.0 及以下)
fastboot flashing unlock	解开 bootloader 锁 (Android 6.0 及以上)
fastboot flashing lock	恢复 bootloader 锁
fastboot flash recovery (filename)	在 bootloader 模式中向设备刷入文件

adb语法

```

> adb --help
Android Debug Bridge version 1.0.41
Version 33.0.2-8557947
Installed as /Users/crifan/dev/dev_tool/android/AndroidSDK/platform-tools/adb

global options:
  -a                      listen on all network interfaces, not just localhost
  -d                      use USB device (error if multiple devices connected)
  -e                      use TCP/IP device (error if multiple TCP/IP devices available)

  -s SERIAL                use device with given serial (overrides $ANDROID_SERIAL)
  -t ID                    use device with given transport id
  -H                      name of adb server host [default localhost]
  -P                      port of adb server [default 5037]
  -L SOCKET                listen on given socket for adb server [default=tcp:localhost:5037]
--one-device SERIAL USB   only allowed with 'start-server' or 'server nodaeon', server
will only connect to one USB device, specified by a serial number or USB device address
s.

--exit-on-write-error     exit if stdout is closed

general commands:
  devices [-l]              list connected devices (-l for long output)
  help                     show this help message
  version                  show version num

networking:
  connect HOST[:PORT]      connect to a device via TCP/IP [default port=5555]
  disconnect [HOST[:PORT]]  disconnect from given TCP/IP device [default port=5555], or all

```

```

pair HOST[:PORT] [PAIRING CODE]
    pair with a device for secure TCP/IP communication
forward --list           list all forward socket connections
forward [--no-rebind] LOCAL REMOTE
    forward socket connection using:
        tcp: port> ( local may be "tcp:0" to pick any open port)
        localabstract: unix domain socket name
        localreserved: unix domain socket name
        localfilesystem: unix domain socket name
        jdwp: process pid> (remote only)
        vsock: CID : port> (remote only)
        acceptfd: fd> (listen only)
forward --remove LOCAL   remove specific forward socket connection
forward --remove-all     remove all forward socket connections
ppp TTY [PARAMETER... ] run PPP over USB
reverse --list           list all reverse socket connections from device
reverse [--no-rebind] REMOTE LOCAL
    reverse socket connection using:
        tcp: port> ( remote may be "tcp:0" to pick any open port)
        localabstract: unix domain socket name
        localreserved: unix domain socket name
        localfilesystem: unix domain socket name
reverse --remove REMOTE  remove specific reverse socket connection
reverse --remove-all     remove all reverse socket connections from device
mdns check               check if mdns discovery is available
mdns services            list all discovered services

file transfer:
push [--sync] [-z ALGORITHM] [-Z] LOCAL... REMOTE
    copy local files/directories to device
    --sync: only push files that are newer on the host than the device
    -n: dry run: push files to device without storing to the filesystem
    -z: enable compression with a specified algorithm (any/none/brotli/lz4/zstd)
    -Z: disable compression
pull [-a] [-z ALGORITHM] [-Z] REMOTE... LOCAL
    copy files/dirs from device
    -a: preserve file timestamp and mode
    -z: enable compression with a specified algorithm (any/none/brotli/lz4/zstd)
    -Z: disable compression
sync [-l] [-z ALGORITHM] [-Z] [all data odm oem product system system_ext|vendor]
    sync a local build from $ANDROID_PRODUCT_OUT to the device (default all)
    -n: dry run: push files to device without storing to the filesystem
    -l: list files that would be copied, but don't copy them
    -z: enable compression with a specified algorithm (any/none/brotli/lz4/zstd)
    -Z: disable compression

shell:
shell [-e ESCAPE] [-n] [-Tt] [-x] [COMMAND...]
    run remote shell command (interactive shell if no command given)
    -e: choose escape character, or "none"; default '~'
    -n: don't read from stdin
    -T: disable pty allocation
    -t: allocate a pty if on a tty (-tt: force pty allocation)
    -x: disable remote exit codes and stdout/stderr separation

```

```

emu COMMAND           run emulator console command

app installation (see also `adb shell cmd package help`):
install [-lrtsdg] [--instant] PACKAGE
    push a single package to the device and install it
install-multiple [-lrtsdpg] [--instant] PACKAGE...
    push multiple APKs to the device for a single package and install them
install-multi-package [-lrtsdpg] [--instant] PACKAGE...
    push one or more packages to the device and install them atomically
    -r: replace existing application
    -t: allow test packages
    -d: allow version code downgrade (debuggable packages only)
    -p: partial application install (install-multiple only)
    -g: grant all runtime permissions
    --abi ABI: override platform's default ABI
    --instant: cause the app to be installed as an ephemeral install app
    --no-streaming: always push APK to device and invoke Package Manager as separate steps
    --streaming: force streaming APK directly into Package Manager
    --fastdeploy: use fast deploy
    --no-fastdeploy: prevent use of fast deploy
    --force-agent: force update of deployment agent when using fast deploy
    --date-check-agent: update deployment agent when local version is newer and using fast deploy
    --version-check-agent: update deployment agent when local version has different version code and using fast deploy
    --local-agent: locate agent files from local source build (instead of SDK location)
)
    (See also `adb shell pm help` for more options.)
uninstall [-k] PACKAGE
    remove this app package from the device
    '-k': keep the data and cache directories

debugging:
bugreport [PATH]
    write bugreport to given PATH [default=bugreport.zip];
    if PATH is a directory, the bug report is saved in that directory.
    devices that don't support zipped bug reports output to stdout.
jdwp                  list pids of processes hosting a JDWP transport
logcat                show device log (logcat --help for more)

security:
disable-verity        disable dm-verity checking on userdebug builds
enable-verity          re-enable dm-verity checking on userdebug builds
keygen FILE           generate adb public/private key; private key stored in FILE,

scripting:
wait-for[-TRANSPORT]-STATE...
    wait for device to be in a given state
    STATE: device, recovery, rescue, sideload, bootloader, or disconnect
    TRANSPORT: usb, local, or any [default any]

```

```

get-state           print offline | bootloader | device
get-serialno       print serial-number
get-devpath        print device-path
remount [-R]
    remount partitions read-write. if a reboot is required, -R will
    will automatically reboot the device.
reboot [bootloader recovery sideload sideload-auto-reboot]
    reboot the device; defaults to booting system image but
    supports bootloader and recovery too. sideload reboots
    into recovery and automatically starts sideload mode,
    sideload-auto-reboot is the same but reboots after sideloading.
sideload OTAPACKAGE   sideload the given full OTA package
root               restart adbd with root permissions
unroot             restart adbd without root permissions
usb                restart adbd listening on USB
tcpip PORT         restart adbd listening on TCP on PORT

internal debugging:
start-server        ensure that there is a server running
kill-server         kill the server if it is running
reconnect           kick connection from host side to force reconnect
reconnect device    kick connection from device side to force reconnect
reconnect offline   reset offline/unauthorized devices to force reconnect

usb:
attach              attach a detached USB device
detach              detach from a USB device to allow use by other processes
environment variables:
$ADB_TRACE          comma-separated list of debug info to log:
                    all,adb,sockets,packets,rwx,usb,sync,sysdeps,transport,jdwp
$ADB_VENDOR_KEYS    colon-separated list of keys (files or directories)
$ANDROID_SERIAL     serial number to connect to (see -s)
$ANDROID_LOG_TAGS   tags to be used by logcat (see logcat --help)
$ADB_LOCAL_TRANSPORT_MAX_PORT max emulator scan port (default 5585, 16 emus)
$ADB_MDNS_AUTO_CONNECT comma-separated list of mdns services to allow auto-connect (
                        default adb-tls-connect)

```

adb文档和资料

- 官网文档
 - [Android 调试桥 \(adb\) | Android 开发者 | Android Developers](#)

The screenshot shows a web browser displaying the URL <https://developer.android.com/studio/command-line/adb>. The page is titled 'Android 调试桥 (adb)' and is part of the '用户指南' (User Guide) section. The left sidebar contains a navigation menu with categories like '发布应用', '发布库', '命令行工具', '概览', 'aapt2', and 'adb'. The 'adb' category is currently selected and highlighted in blue. The main content area provides an overview of the adb command-line interface, its components (Client, Server, Daemon), and how to connect devices via Wi-Fi or USB. A sidebar on the right lists various adb commands and their descriptions.

- 其他资料
 - [Home | Android Debug Bridge \(ADB\) Commands Manual \(adbcommand.com\)](#)
- Listing of adb Commands
 - [Android Debug Bridge | Android Developers \(oschina.net\)](#)

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：
2022-11-10 14:49:22

相关心得

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 22:10:51

文件管理

TODO:

- 【记录】从Nox夜神模拟器中拷贝文件到Mac中
 -
 - 【已解决】安卓手机Google Pixel3用文件管理管理文件
 - 【已解决】安卓手机中打开log日志文件查看内容
-

安卓逆向期间，常涉及到文件管理方面的内容。

文件上传和下载=导入导出文件

- 用文件管理器
- 用命令：adb
 - 导出文件

```
adb pull /mnt/sdcard/Download/com.lanyou.bydwj.ikk/
```

- 导入文件

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

```
adb push riru-v25.4.4-debug.zip /sdcard/Download
```

文件目录的关系

常见目录：

- /sdcard : SD卡根目录，即普通用户保存文件的根目录
 - /sdcard/Download : 下载目录
 - 安卓手机中，通过 浏览器 等工具下载的文件，往往默认保存在到 下载 目录，就是这个： /sdcard/Download

对应的查看效果：

- 文件管理器
 - 文件极客
 -

3:39



← 下载



全部

Download

今天



EdXposed-v0.5.2.2_4683-master-z...



14.79 MB, 9分钟前



MagiskHidePropsConf-v6.1.2.zip



97.97 KB, 5小时前

昨天



magisk_patched-25200_lOhGo.img



67.11 MB, 22小时前



magisk_log_2022-09-14T17.12.35...



44.83 KB, 22小时前

8月 31, 周三



com.google.android.youtube_16.2...



101 MB, 8月31日

2009 1月 01



boot.img

67.11 MB 2009年1月1日



- 命令行
 - adb shell

```
crifan@licrifandeMacBook-Pro ~ /dev/dev_tool/android/EdXposed$ adb shell
blueLine:/ $ whoami
shell
blueLine:/ $ cd /sdcard/Download
blueLine:/sdcard/Download $ ls -lh
total 120M
-rw-rw---- 1 root everybody 14M 2022-09-15 15:30 EdXposed-v0.5.2.2_4683-master-
z-debug.zip
-rw-rw---- 1 root everybody 96K 2022-09-15 10:02 MagiskHidePropsConf-v6.1.2.zip
-rw-rw---- 1 root everybody 64M 2009-01-01 00:00 boot.img
-rw-rw---- 1 root everybody 97M 2022-08-31 11:16 com.google.android.youtube_16.
29.36.apk
-rw-rw---- 1 root everybody 44K 2022-09-14 17:12 magisk_log_2022-09-14T17.12.35
.log
-rw-rw---- 1 root everybody 64M 2022-09-14 17:18 magisk_patched-25200_10hGo.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:
2022-11-10 14:49:53

文件管理器

TODO:

- 【已解决】安卓手机Google Pixel3用文件管理管理文件
 - 【已解决】安卓手机中打开log日志文件查看内容
 - 【部分解决】安卓手机Google Pixel3中用RE文件管理器拷贝和移动文件夹
 - 【已解决】导出安卓手机Google Pixel3中的文件
 - 【未解决】安卓手机Google Pixel3中用ES文件管理器拷贝和移动文件夹
 - apk安装
 - 【已解决】安卓手机中安装YouTube的apk报错：应用未安装
 - 【已解决】安卓中把微信传输收到的apk.1改名为apk去掉.1的后缀
 - 【无需解决】Android 9的Google Pixel3中安装apk报错应用未安装：可能和sdkVersion有关
 - 【已解决】安卓手机Google Pixel3中用ES文件管理器安装youtube的apk
 - 【已解决】Google Pixel3中如何彻底卸载提示未针对此用户安装的YouTube应用
 - 【已解决】安卓手机Google Pixel3中安装apk应用
 - 【未解决】用腾讯管家的APK安装器安装安卓apk应用
-

- Android的文件管理器
 - 推荐： 文件管理
 - 应用宝中搜 文件管理， 而找到的
 -

5:29



99%



文件管理 官方

文件文件管理器文件压缩文件解压

文件服务

文件管理

3.8

82人评分

3255.8万

下载量

详情评价¹²

文件智能分类

手机文件智能分类便于管理



文件编辑管理

手机所有文件多种编辑状态



微信文件管理

微信及QQ的文件分类管理



文档

手机文

软件介绍

版本号 1.2.2

文件、文件管理器、文件分类、文件压缩、文件解压

文本，图片，音乐，视频，多功能与一身。
移动、排序、新建、备份，都能满足你。

文件●功能齐全，涵盖全面的文件操作

46.4%



- 其他可选
 - ES文件管理器
 - RE
 - MT管理器
 - 安卓模拟器中的文件管理器
 - 夜神模拟器中自带文件浏览器
- 常见的用途
 - 给文件改名
 - xxx.apk.1 -> xxx.apk
 - 查找文件
 - 再去安装apk文件
 - 查看文件内容
 - 查看log日志内容

MT管理器

- MT = MT管理器 = MT Manager = MT Manager for Android
- 是什么：安卓中的一个文件管理器
- 常被简称为： MT2
 - 因为最新版本是v2
 - 比如：
 - MT浏览器_V2.5.4.apk 文件管理器
 - 文件管理神器 MT Manager v2.6.1 for Android
 - MT管理器2.0

特点

- 支持在VirtualXposed中使用MT
- 除了普通文件管理功能外，还支持APK反编译相关功能

应用简介

MT管理器是一款强大的文件管理工具和APK逆向修改神器。

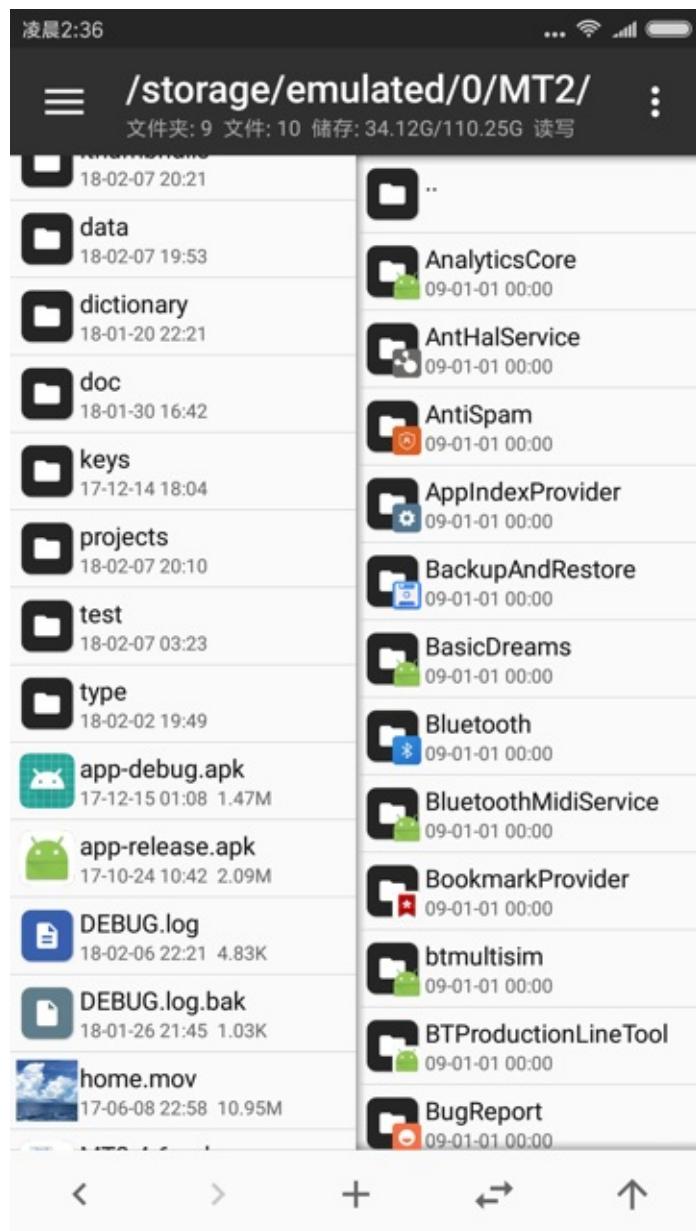
- 如果你喜欢它的双窗口操作风格，可以单纯地把它当成文件管理器使用。
- 如果你对修改APK有浓厚的兴趣，那么你可以用它做许许多多的事
 - 例如汉化应用、替换资源、修改布局、修改逻辑代码、资源混淆、去除签名校验等，主要取决于你如何使用。

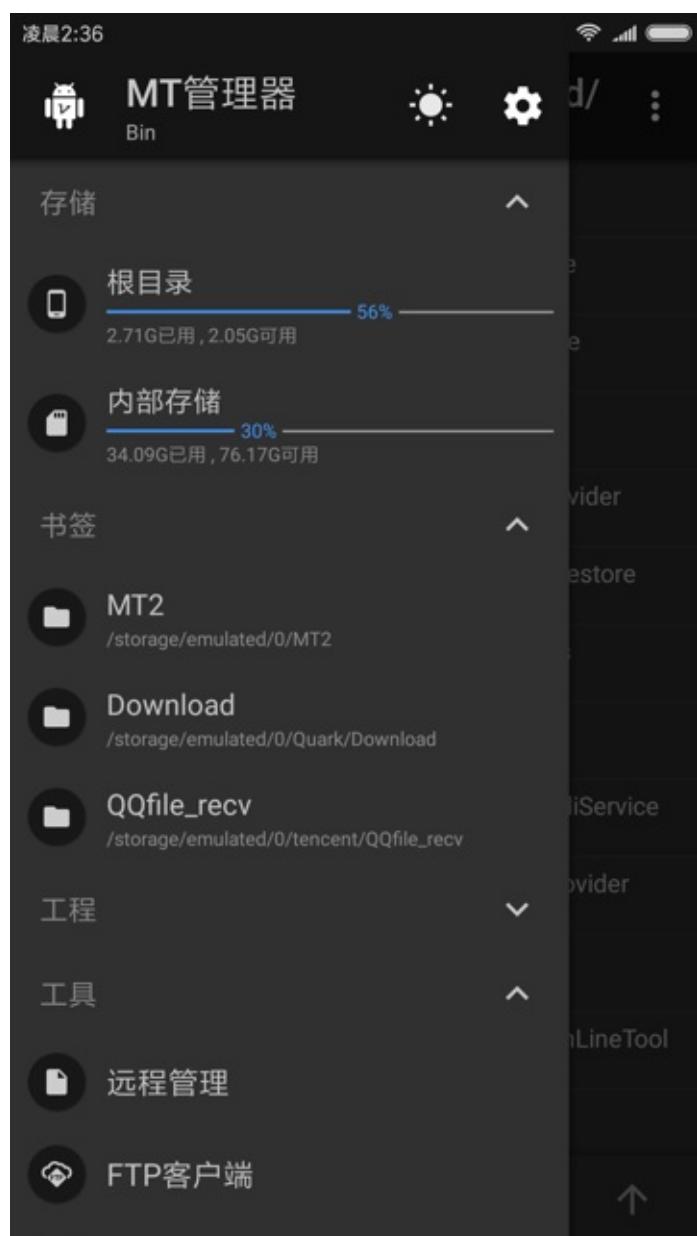
主要功能

- 文件复制、移动、创建软链接、重命名、删除、创建文件(夹)，文件批量操作。获取 Root 权限后可访问系统目录，挂载文件系统为读写，修改文件权限和所有者。
- 像 WinRAR 那样打开 ZIP 格式文件，可以对 ZIP 内的文件进行删除、重命名、移动，添加/替换外部文件到 ZIP 中，无需解压后再重新打包，同时支持单独解压 ZIP 内的部分文件。
- 自带强大的文本编辑器，可以流畅编辑大文本文件，支持设置是否显示行号、开关自动换行、双指缩

- 放字体大小、自动识别编码、代码语法高亮、自动缩进、正则搜索替换。
- 拥有图片查看、音乐播放、字体预览、执行脚本、文本对比等功能，在侧拉栏中可方便地查看存储设备、FTP连接、书签、后台、工具等。
 - APK 编辑功能，主要有 DEX 编辑，ARSC 编辑，XML 编辑，APK 签名、APK 优化、APK 共存、去除签名校验、RES 资源混淆、RES 反资源混淆、翻译模式等。

截图举例









AndroidManifest.xml
Android Xml (反编译) *

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.
3   android.com/apk/res/android"
4     package="cn.edu.njupt.cengceng"
5     platformBuildVersionCode="25"
6     platformBuildVersionName="7.1.1"
7     android:versionCode="1"
8     android:versionName="1.0">
9       <uses-sdk
10         android:minSdkVersion="14"
11         android:targetSdkVersion="25" />
12       <!-- 拥有完全的网络访问权限 -->
13       <uses-permission android:name="android.
14         permission.INTERNET" />
15       <application
16         android:theme="@style/AppTheme"
17         android:label="@string/app_name"
18         android:icon="@mipmap/ic_launcher"
19         android:allowBackup="true"
20         android:supportsRtl="true"
21         android:roundIcon="@mipmap/
22           ic_launcher_round">
23           <activity
24             android:theme=
25               <NoActionBar>
26                 android:label=
27                 android:name=
```

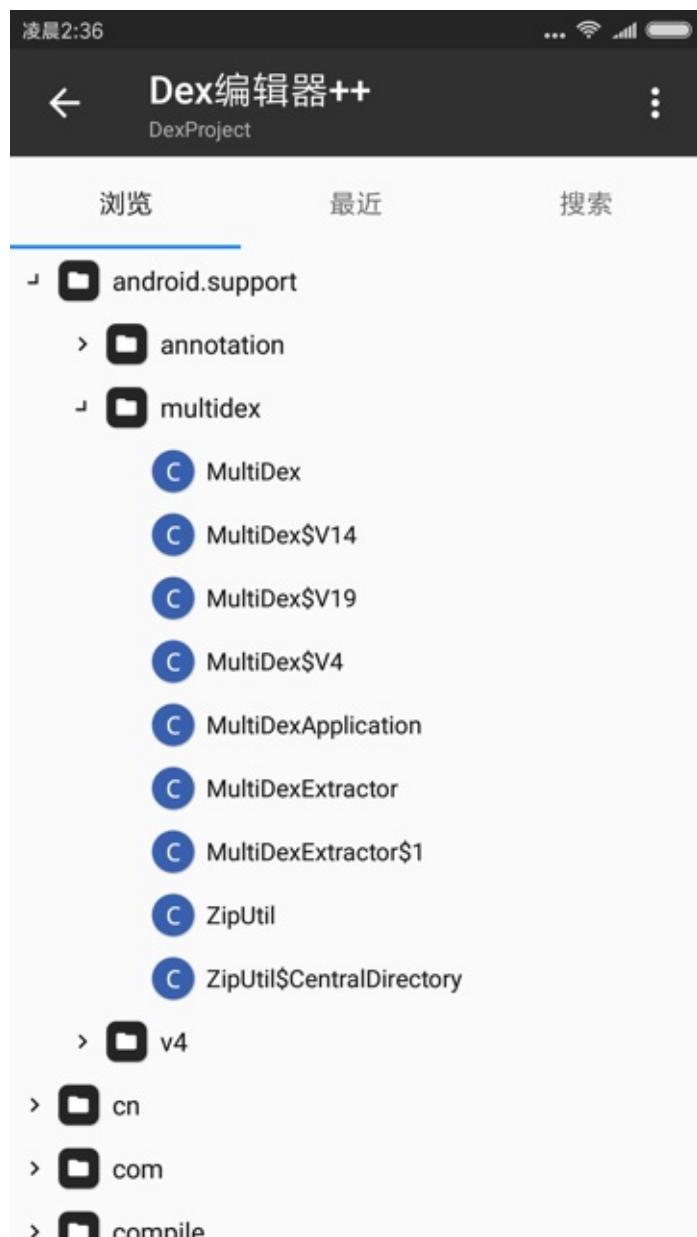
正则表达式

区分大小写

查找 android

关闭

上个 下个 替换 全部 :



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:
2022-11-10 14:51:08

相关资料

- DDMS
 - [Using Dalvik Debug Monitor Service \(DDMS\) | Android Developers \(sourceforge.net\)](#)
 - [Using DDMS | Android Developers \(android-doc.github.io\)](#)
- 常见安卓模拟器的调试端口

- - <https://developer.android.com>
 - 安卓 调试 debug
 - [调试应用 | Android 开发者 | Android Developers](#)

- Android Device Monitor
 - [Android Device Monitor | Android 开发者 | Android Developers](#)

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:
2023-09-13 21:30:01

Android API Level

Android的 API Level 、 sdkVersion

[API Levels | Android versions, SDK/API levels, version codes, codenames, and cumulative usage](https://apilevels.com/)

The screenshot shows a dark-themed web page with a header "Android API Levels". Below it is a note: "This is an overview of all Android versions and their corresponding identifiers for Android developers. Anyone is welcome to open an issue or pull request. Happy developing!" A navigation bar at the top includes "Definitions", "Footnotes", and "See also". The main content is a table with the following columns: Version, SDK / API level, Version code, Codename, Cumulative usage¹, and Year.

Version	SDK / API level	Version code	Codename	Cumulative usage ¹	Year
Android 13	Level 33	TIRAMISU	Tiramisu ²	No data	2022
Android 12	Level 32 <small>Android 12L</small>	S_V2	Snow Cone ²	20.7%	2021
	Level 31 <small>Android 12</small>	S			
	<ul style="list-style-type: none"> ▪ targetSdk must be 31+ for new apps. ▪ targetSdk will need to be 31+ for app updates by Nov 2022 and all existing apps by Nov 2023.³ 				
Android 11	Level 30	R	Red Velvet Cake ²	50.3%	2020
	<ul style="list-style-type: none"> ▪ targetSdk must be 30+ for app updates, and new WearOS apps. ▪ targetSdk will need to be 30+ for all existing apps by November 2022.³ 				
Android 10	Level 29	Q	Quince Tart ²	72.1%	2019
Android 9	Level 28	P	Pie	82.9%	2018
	<ul style="list-style-type: none"> ▪ targetSdk must be 28+ for Wear OS app updates. 				
Android 8	Level 27 <small>Android 8.1</small>	O_MR1	Oreo	88.4%	2017
	Level 26 <small>Android 8.0</small>	O		91.1%	
Android 7	Level 25 <small>Android 7.1</small>	N_MR1	Nougat	92.5%	2016
	Level 24 <small>Android 7.0</small>	N		95.1%	
Android 6	Level 23	M	Marshmallow	97.4%	2015
Android 5	Level 22 <small>Android 5.1</small>	LOLLIPOP_MR1	Lollipop	98.8%	2015
	Level 21 <small>Android 5.0</small>	LOLLIPOP_L		No data	
	<ul style="list-style-type: none"> ▪ Jetpack Compose requires a minSdk of 21 or higher. 				
Android 4	Level 20 <small>Android 4.4W</small> ⁴	KITKAT_WATCH	KitKat		

https://apilevels.com

Android 4	Level 20 Android 4.4W ⁴	KITKAT_WATCH	KitKat	2013	
	Level 19 Android 4.4	KITKAT			
	▪ Google Play services do not support Android versions below API level 19.				
	Level 18 Android 4.3	JELLY_BEAN_MR2	Jelly Bean		
	Level 17 Android 4.2	JELLY_BEAN_MR1			
	Level 16 Android 4.1	JELLY_BEAN			
	Level 15 Android 4.0.3 – 4.0.4	ICE_CREAM_SANDWICH_MR1	Ice Cream Sandwich		
	Level 14 Android 4.0.1 – 4.0.2	ICE_CREAM_SANDWICH			
	▪ Jetpack/AndroidX libraries require a <code>minSdk</code> of 14 or higher.				
	Android 3		Honeycomb		
Level 13 Android 3.2		HONEYCOMB_MR2			
Level 12 Android 3.1		HONEYCOMB_MR1			
Level 11 Android 3.0		HONEYCOMB			
Android 2		GINGERBREAD_MR1	Gingerbread	2010	
Level 10 Android 2.3.3 – 2.3.7		GINGERBREAD			
Level 9 Android 2.3.0 – 2.3.2		FROYO			
Level 8 Android 2.2		ECLAIR_MR1	Eclair		
Level 7 Android 2.1		ECLAIR_0_1			
Level 6 Android 2.0.1		ECLAIR			
Android 1		DONUT	Donut	2009	
Level 4 Android 1.6		CUPCAKE	Cupcake		
Level 3 Android 1.5		BASE_1_1	Petit Four		
Level 1 Android 1.0		BASE	None		

Definitions

Gradle files

Kotlin variable	Groovy variable	Definition
<code>minSdk</code>	<code>minSdkVersion</code>	The minimum SDK version your app will support, defined in <code>build.gradle</code> . For

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 22:11:44

子教程

- 父教程
 - [安卓应用的安全和破解](#)
- 子教程
 - 逆向
 - [Android逆向开发](#)
 - [Android逆向：开启root](#)
 - [Android逆向：静态分析](#)
 - [安卓逆向利器：JEB](#)
 - [安卓反编译利器：jad](#)
 - [Android逆向：动态调试](#)
 - [Android逆向：重新打包apk](#)
 - 相关
 - [安卓模拟器](#)
 - [好用的安卓模拟器：夜神Nox](#)
 - 框架
 - [安卓逆向调试：XPosed框架](#)
 - 代码
 - [crifan/AndroidYouTubeDynamicDebug: 安卓逆向动态调试YouTube](#)

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：
2023-10-04 16:28:30

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：
2022-10-27 16:42:54

参考资料

- YouTube 安卓逆向 逆向思路
- 【整理】安卓逆向：相关资料
- 【整理】安卓逆向：开发心得
- 【整理】adb命令语法帮助信息
-
- [Android逆向：重新打包apk](#)
- [Android逆向：静态分析](#)
-
- [Creating a URL \(The Java™ Tutorials > Custom Networking > Working with URLs\) \(oracle.com\)](#)
- [Java URL class- javatpoint](#)
- [A Simple Guide to the Java URL | Baeldung](#)
- [java - Create URL from a String - Stack Overflow](#)
- [Android逆向破解：使用Android Studio调试反编译后的smali代码 - 简书 \(jianshu.com\)](#)
- [什么刷机软件最好用\(oppo手机用什么刷机软件好用\) - 52FMZ资讯网](#)
-

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：
2023-08-30 20:57:07