目录

Android逆向开发

- 最新版本：`v0.6`
- 更新时间：`20221109`

简介

介绍Android逆向开发相关的内容。包括逆向的核心思路；以及Hook插件与开发，包括Xposed、Cydia Substrace，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

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

概览

此处介绍安卓的安全和逆向中的安卓的逆向中关于逆向开发相关的内容。

逆向核心思路

TODO：

【未解决】Android的YouTube逆向：研究request请求api和protobuf相关部分代码

此处介绍安卓逆向的核心思路：

- 找核心代码的入口点
  - 对于Android来说，往往是 `java` 中相关的 `基础的类` ， `内置的类`
  - 比如
    - `网络请求` 类
    - `url` 相关的类
      - 比如
        - `new URL`
    - `字符串` 相关的类
      - 可能会用到URL拼接，其内部涉及到 `字符串` 的拼接
      - 比如
        - `append`
        - `getBytes`
        - `String builder`
- 再去写hook代码，加过滤条件，用工具调试
  - `Xposed` ：用 `hook` 框架去写hook代码
  - `frida` ：调试逻辑
- 期间配合抓包
  - 先要 `抓包` 找相关 `url`
    - 后续才知道要过滤哪些url

Java基础的内置的类

URL

```
URL myURL = new URL("http://example.com/pages/");
URL page1URL = new URL(myURL, "page1.html");
URL page2URL = new URL(myURL, "page2.html");
```

或：

```
new URL("http", "example.com", "/pages/page1.html");
```

或：

```
//URLDemo.java
```

```
import          *;

public class URLDemo {

  public static void main(String[] args) {
    try {
      URL url = new URL("http://www.javatpoint.com/java-tutorial");

      System.out.println("Protocol: " + url.getProtocol());
      System.out.println("Host Name: " + url.getHost());
      System.out.println("Port Number: " + url.getPort());
      System.out.println("File Name: " + url.getFile());
    } catch (Exception e) {
      System.out.println(e);
    }
  }
}
```

或:

```
import          *;

URL url = new URL("/a-guide-to-java-sockets");

URL home = new URL("http://baeldung.com");
URL url = new URL(home, "a-guide-to-java-sockets");
```

或:

```
@Test
public void givenBaseUrl_whenCreatesRelativeUrl_thenCorrect() {
    URL baseUrl = new URL("http://baeldung.com");
    URL relativeUrl = new URL(baseUrl, "a-guide-to-java-sockets");

    assertEquals("http://baeldung.com/a-guide-to-java-sockets",
      relativeUrl.toString());
}
```

或:

```
URL url = new URL(yourUrl, "/api/v1/status.xml");
```

或:

```
URL domain = new URL("http://example.com");
URL url = new URL(domain + "/files/resource.xml");
```

hook插件与开发

TODO：

- 把如何用XPosed写Android的hook插件的帖子整理过来
-
- 【已解决】Android 11的Magisk中安装EdXposed 0.5.2.2结果失败：请先从Magisk Manager中安装Riru Installation failed
- 【已解决】在Android 11中安装EdXposed框架

- Android的Hook插件开发框架
  - `Xposed`
  - `Cydia Substrate`

Xposed

- `Xposed`
  - 典型用途
    - 用于安装相关插件比如 `FDex2` ，配合安卓破解，导出运行时 `app` 的 `dex` 文件
    - 基于 `XPosed` 编写hook插件，实现特定功能
  - 使用举例
    - 安装到 `夜神模拟器` 中的 `XPosed`
      

  - 最新为： `EdXposed`
  - 详见

- 独立教程：强大的安卓破解辅助工具：XPosed框架

Cydia Substrate

- 主页
  - Cydia Substrate
- 下载
  - com.saurik.substrate.apk
- 功能
  - 和 `Xposed` 类似的框架，用来安装各种插件，实现各种功能。
    - 比如可以：
      - 安装绕过 `ssl` 检测的插件，用来破解 `ssl pinning`
        - 关于安卓的app中的https：
          - app内部启用了：
            - `SSL Pinning` = `ssl certificate pinning` = `certificate pinning`
              - = `ssl证书绑定` =证书绑定`
      - 此处也可以用来安装相关插件，导出安卓的dex文件
- 特点
  - Hook底层方法非常方便
    - 对so中的方法hook操作非常便捷
- 截图
  -

Android-OpenDebug

- 主页
  - iSECPartners/Android-OpenDebug: Make any application debuggable
- 功能
  - 是一个 `Cydia Substrate` 的插件
    - 所以前提是要先安装 `Cydia Substrate`
  - 可以使得任何一个安卓程序可以被调试
    - 就有了分析和破解的可能
- 下载
  - Android-OpenDebug APK下载
- 安装
  - `adb install Android-OpenDebug.apk`
  - 或直接安装apk

Introspy-Android

- 主页
  - GitHub
    - iSECPartners/Introspy-Android: Security profiling for blackbox Android
  - 网站
    - Introspy-Android
- 功能
  - 帮助分析安卓app运行期间的行为
    - 以便于找到可能存在的安全问题
- 提示
  - 是个 `Cydia Substrate` 插件
    - 所以前提是先安装 `Cydia Substrate`

其他心得

adb

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
  - `#` = 井号 ： `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

- 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 nodaemon', server will on
ly connect to one USB device, specified by a serial number or USB device address.
 --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
```

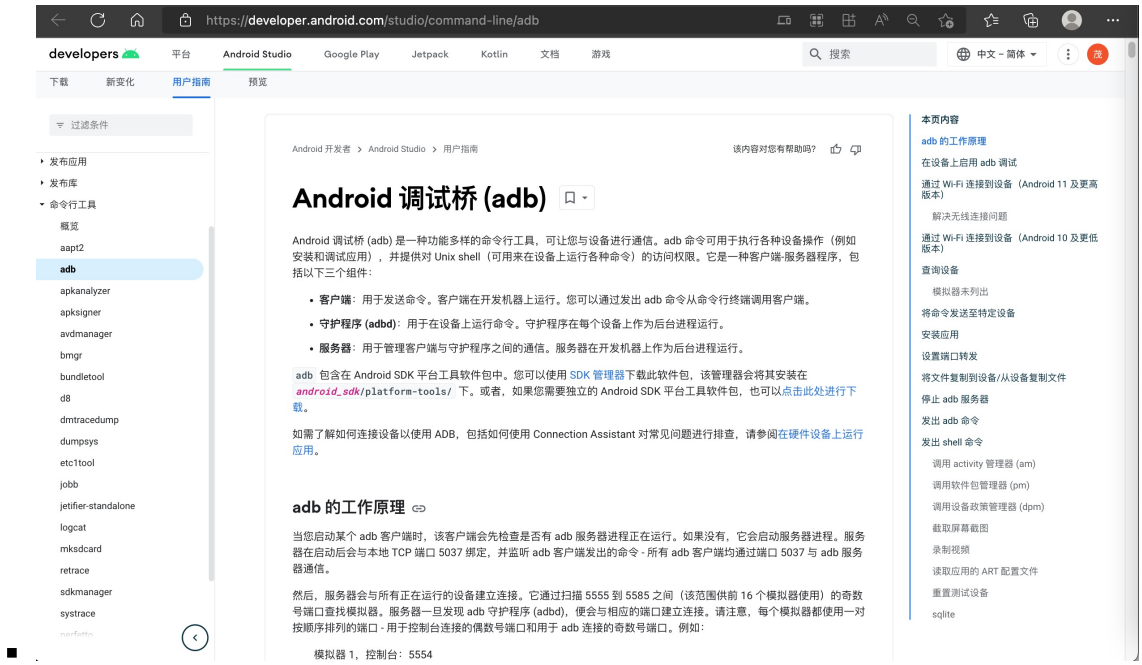
```
file transfer:
 push [--sync] [-z ALGORITHM] [-Z] LOCAL... REMOTE
     copy local files/directories to device
     --sync: only push files that are newer on the host than the device
     -n: dry run: push files to device without storing to the filesystem
     -z: enable compression with a specified 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 dep
loy
     --version-check-agent: update deployment agent when local version has different version co
de 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



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

文件管理

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

- 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.ap
k
-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_lOhGo.img
```

文件管理器

TODO：

- 【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
- 【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Google Pixel3中用ES文件管理器安装youtube的apk
- 【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
-

- 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
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有深厚的兴趣，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。

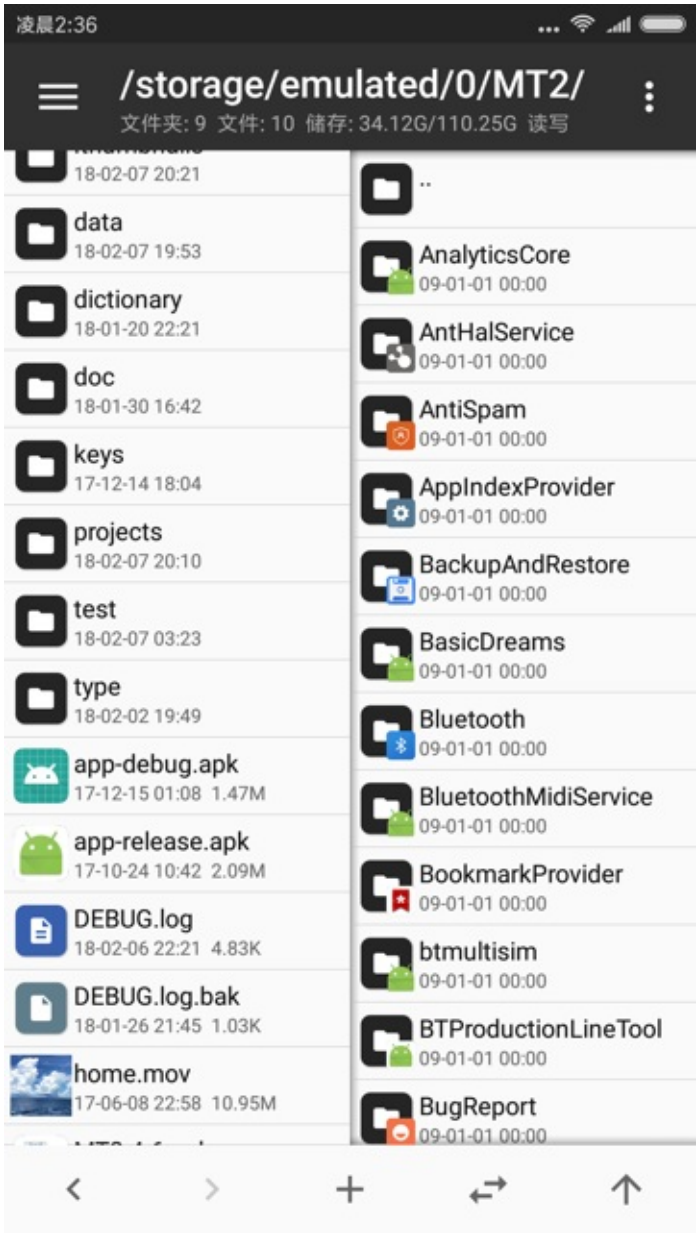
主要功能

- 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。

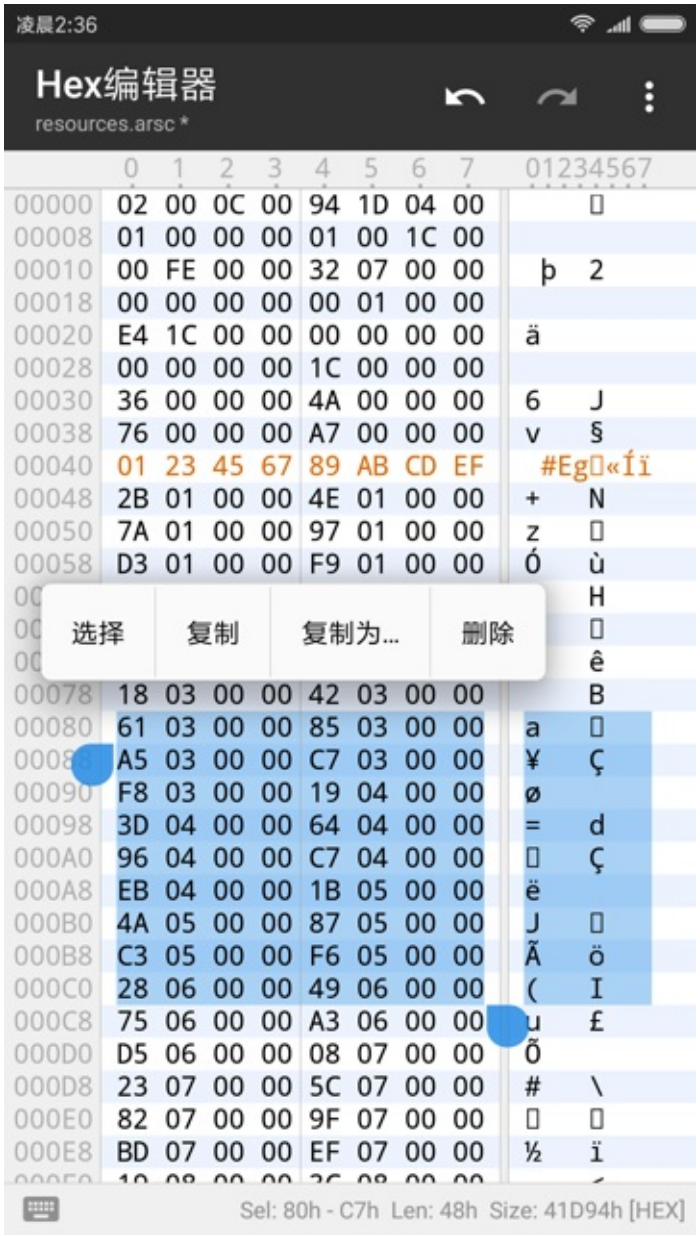
- 拥有图片查看、音乐播放、字体预览、执行脚本、文本对比等功能，在侧拉栏中可方便地查看存储设备、FTP连接、书签、后台、工具等。
- APK 编辑功能，主要有 DEX 编辑，ARSC 编辑，XML 编辑，APK 签名、APK 优化、APK 共存、去除签名校验、RES 资源混淆、RES 反资源混淆、翻译模式等。

截图举例

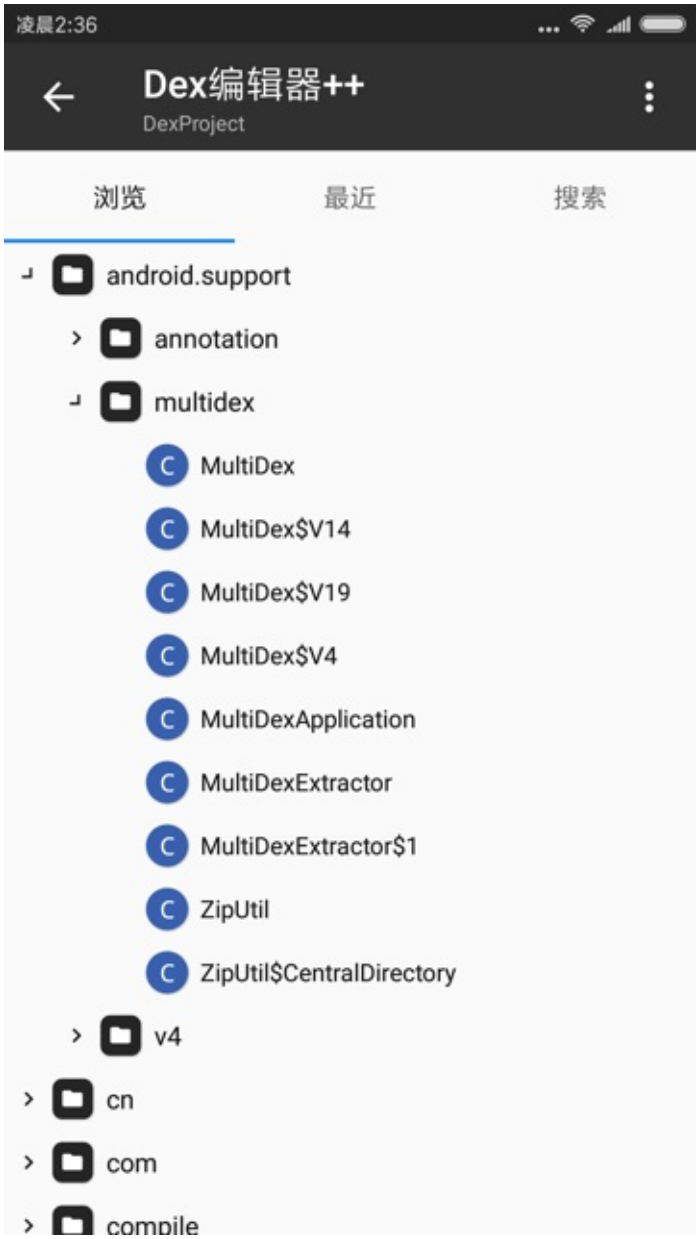



凌晨2:36
MT管理器
Bin
存储
根目录
56%
2.71G已用，2.05G可用
内部存储
30%
34.09G已用，76.17G可用
书签
MT2
/storage/emulated/0/MT2
Download
/storage/emulated/0/Quark/Download
QQfile_recv
/storage/emulated/0/tencent/QQfile_recv
工程
工具
远程管理
FTP客户端


凌晨2:36
/storage/emulated/0/MT2/
文件夹: 9 文件: 10 储存: 34.12G/110.25G 读写
.thumbnails
18-02-07 20:21
..
data
18-02-07 19:53
AnalyticsCore
09-01-01 00:00
dictionary
18-01-20 22:21
AntHalService
09-01-01 00:00
doc
AntiSpam
复制 ->
移动 ->
链接 ->
重命名
删除
压缩
属性
分享
打开方式...
添加书签
17-10-24 10:42 2.09M
09-01-01 00:00
DEBUG.log
18-02-06 22:21 4.83K
btmultisim
09-01-01 00:00
DEBUG.log.bak
18-01-26 21:45 1.03K
BTProductionLineTool
09-01-01 00:00
home.mov
17-06-08 22:58 10.95M
BugReport
09-01-01 00:00


凌晨2:36
Hex编辑器
resources.arsc *
选择
复制
复制为...
删除
Sel: 80h - C7h Len: 48h Size: 41D94h [HEX]


凌晨2:36
AndroidManifest.xml
Android Xml (反编译) *
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.
android.com/apk/res/android"
package="cn.edu.njupt.cengceng"
platformBuildVersionCode="25"
platformBuildVersionName="7.1.1"
android:versionCode="1"
android:versionName="1.0">
<uses-sdk
android:minSdkVersion="14"
android:targetSdkVersion="25" />
<!-- 拥有完全的网络访问权限 -->
<uses-permission android:name="android.
permission.INTERNET" />
<application
android:theme="@style/AppTheme"
android:label="@string/app_name"
android:icon="@mipmap/ic_launcher"
android:allowBackup="true"
android:supportsRtl="true"
android:roundIcon="@mipmap/
ic_launcher_round">
<activity
android:theme
NoActionBar"
android:label=
android:name
正则表达式
区分大小写
关闭
查找 android
上个
下个
替换
全部



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

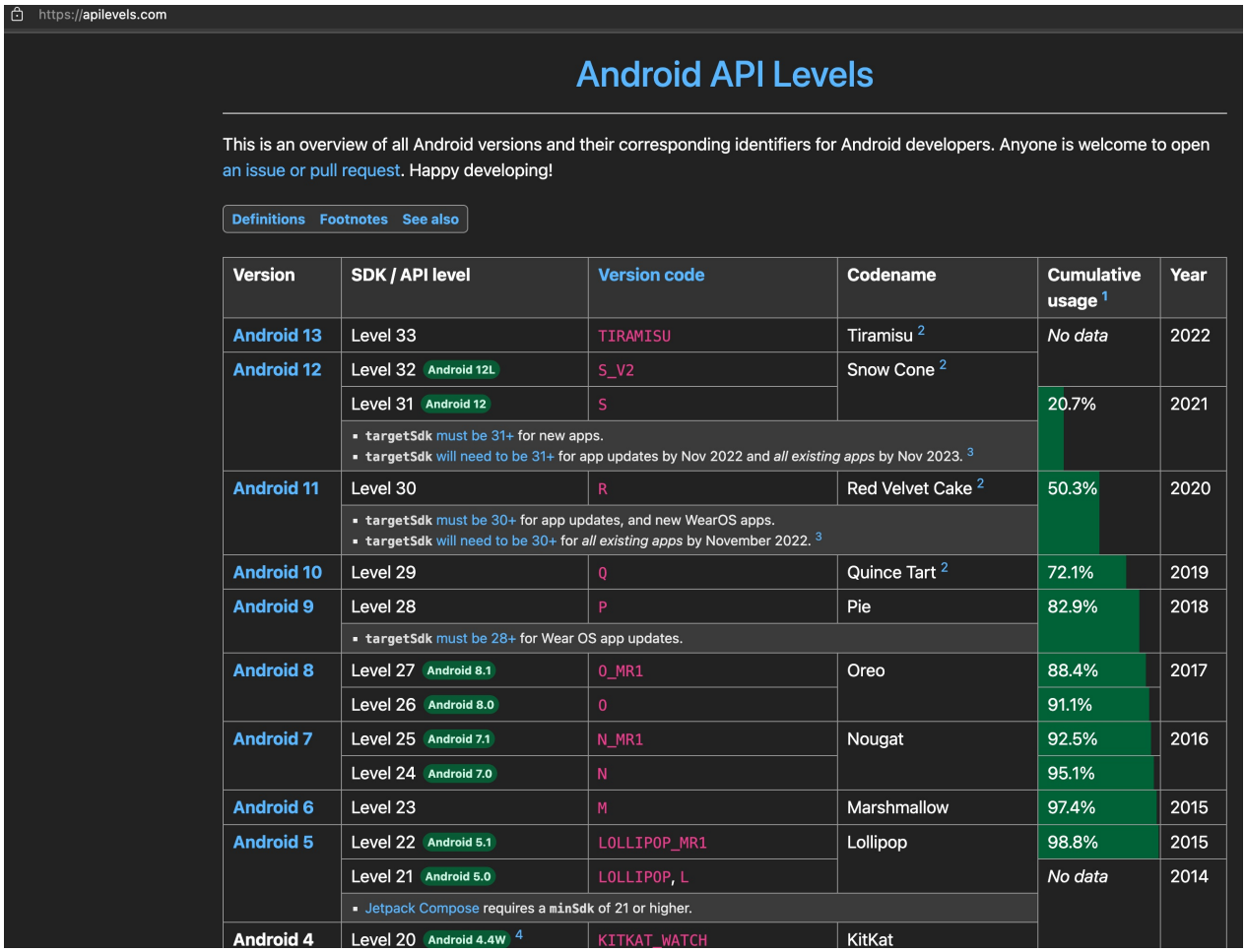
- https://developer.android.com
  - 安卓 调试 debug
    - 调试应用 | Android 开发者 | Android Developers

- Android Device Monitor
  - Android Device Monitor | Android 开发者 | Android Developers

Android API Level

Android的 `API Level` 、 `sdkVersion`

API Levels | Android versions, SDK/API levels, version codes, codenames, and cumulative usage



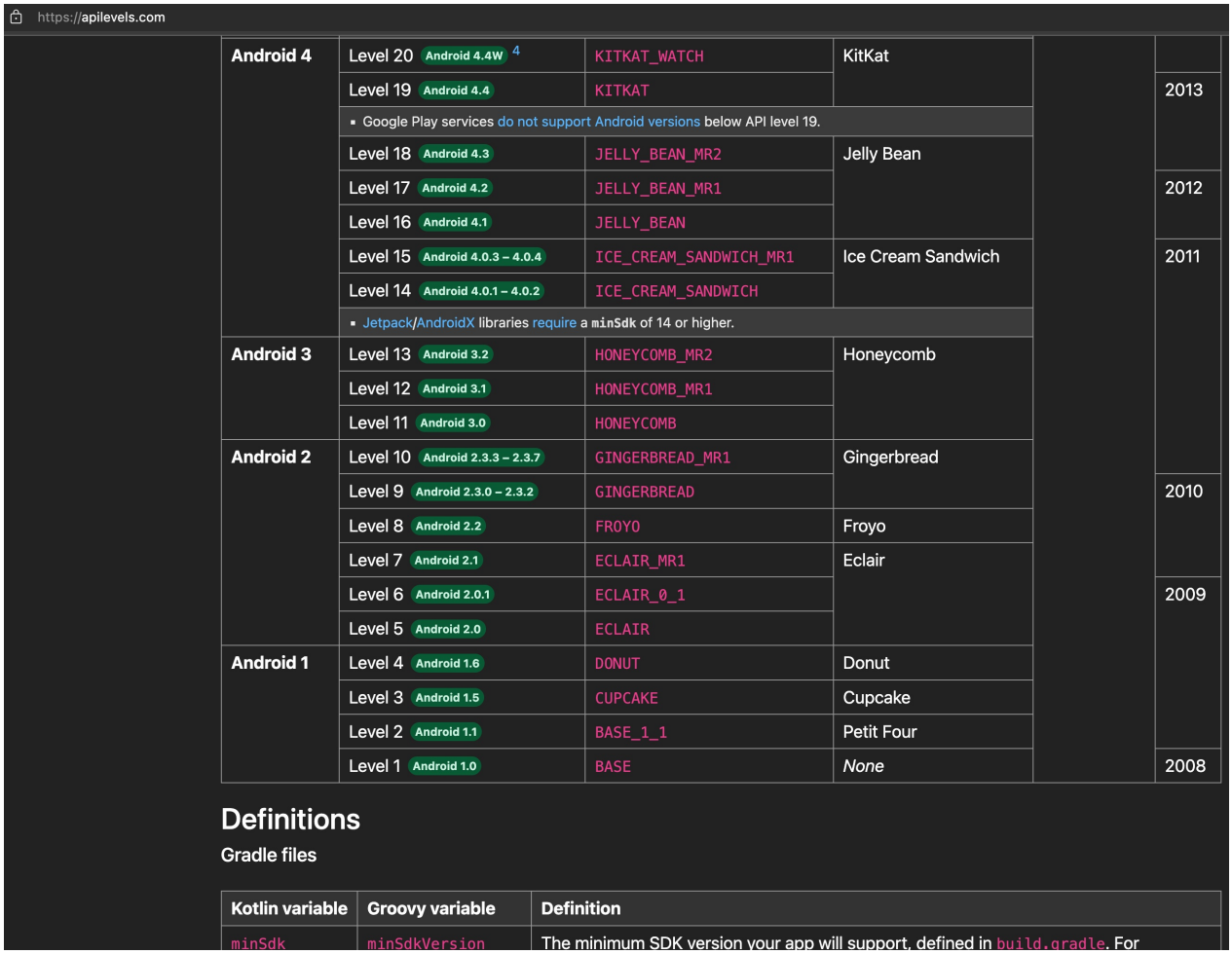
https://apilevels.com

Android API Levels

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!

Definitions Footnotes See also

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 Android 12L	S_V2	Snow Cone [2]		
	Level 31 Android 12	S		20.7%	2021
	▪ `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
	▪ `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
	▪ `targetSdk` must be 28+ for Wear OS app updates.				
Android 8	Level 27 Android 8.1	O_MR1	Oreo	88.4%	2017
	Level 26 Android 8.0	O		91.1%	
Android 7	Level 25 Android 7.1	N_MR1	Nougat	92.5%	2016
	Level 24 Android 7.0	N		95.1%	
Android 6	Level 23	M	Marshmallow	97.4%	2015
Android 5	Level 22 Android 5.1	LOLLIPOP_MR1	Lollipop	98.8%	2015
	Level 21 Android 5.0	LOLLIPOP, L		*No data*	2014
	▪ Jetpack Compose requires a `minSdk` of 21 or higher.				
Android 4	Level 20 Android 4.4W [4]	KITKAT_WATCH	KitKat		



https://apilevels.com

Android 4	Level 20 Android 4.4W [4]	KITKAT_WATCH	KitKat		
	Level 19 Android 4.4	KITKAT			2013
	▪ 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			2012
	Level 16 Android 4.1	JELLY_BEAN			
	Level 15 Android 4.0.3 – 4.0.4	ICE_CREAM_SANDWICH_MR1	Ice Cream Sandwich		2011
	Level 14 Android 4.0.1 – 4.0.2	ICE_CREAM_SANDWICH			
	▪ Jetpack/AndroidX libraries require a **minSdk** of 14 or higher.				
Android 3	Level 13 Android 3.2	HONEYCOMB_MR2	Honeycomb		
	Level 12 Android 3.1	HONEYCOMB_MR1			
	Level 11 Android 3.0	HONEYCOMB			
Android 2	Level 10 Android 2.3.3 – 2.3.7	GINGERBREAD_MR1	Gingerbread		
	Level 9 Android 2.3.0 – 2.3.2	GINGERBREAD			2010
	Level 8 Android 2.2	FROYO	Froyo		
	Level 7 Android 2.1	ECLAIR_MR1	Eclair		
	Level 6 Android 2.0.1	ECLAIR_0_1			2009
	Level 5 Android 2.0	ECLAIR			
Android 1	Level 4 Android 1.6	DONUT	Donut		
	Level 3 Android 1.5	CUPCAKE	Cupcake		
	Level 2 Android 1.1	BASE_1_1	Petit Four		
	Level 1 Android 1.0	BASE	*None*		2008

Definitions

Gradle files

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

子教程

TODO：

- 【未解决】给OPPO R11s刷第三方Recovery：TWRP
- 【整理】安卓root工具：Magisk
- 【已解决】给Android 11的Google Pixel3去开启root权限

- 安卓逆向开发的子教程
  - 逆向调试
    - 概述
      - 主要是用 `Android Studio` 去调试 `apk` 导出的 `smali` 代码
      - 辅助用 `frida` 配合调试逻辑
    - 详见
      - Android逆向：动态调试 (crifan.org)
  - Android逆向：重新打包apk (crifan.org)
  - Android逆向：开启root (crifan.org)

附录

下面列出相关参考资料。

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
-
- 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
-